Survey and Implementation of Commercial Manual Controllers for a Generic Telerobotics Architecture

THESIS

Thomas E. Deeter
Captain, USAF

AFIT/GE/ENG/94D

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/94D-04

Survey and Implementation of Commercial Manual Controllers for a Generic Telerobotics Architecture

THESIS

Thomas E. Deeter
Captain, USAF

AFIT/GE/ENG/94D

Approved for public release; distribution unlimited

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

AFIT/GE/ENG/94D-04

SURVEY AND IMPLEMENTATION OF COMMERCIAL MANUAL CONTROLLERS FOR A GENERIC TELEROBOTICS ARCHITECTURE

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Thomas E. Deeter, BSEE

Captain, USAF

DECEMBER, 1994

Approved for public release; distribution unlimited

ACKNOWLEDGMENTS

I would like to extend my deepest appreciation to my wife and children. Even though my wife experienced a great loss, went through major surgery, and raised our twins and 5 year old on her own, she never laid her burdens on me and she always had a nice word to say when things were going bad; thus giving me the ability to accomplish this research. I would also like to thank my thesis advisor, Maj. Dean Schneider, for his unending patience during this seemingly unending thesis work. My gratitude goes to my thesis committee, Dr. Curtis Spenny and Capt. Keith Shomper, who had to endure my creative writing. My many thanks goes to the guys behind the scenes at AFIT such as Dave Doak, Dan Zambon, and Gary Mauersberger. They always kept the computers operating even through the students (me included) inventive use of the systems.

To Lt Col. Wailes, I say thank you for lending your equipment to the robotics laboratory. Without the logic analyzer, this thesis would have taken many many more hours and would have cost the Air Force a considerable amount of money.

Table of Contents

Page

ACKNOWLEDGMENTS ii

List of Figures iv

List of Tables v

ABSTRACT vi

I. Introduction 1-1
- I.1 Motivation 1-1
- I.2 Scope 1-3
- I.3 Overview 1-4
- I.4 Equipment Required 1-5
- I.5 Conclusions 1-6

II. Previous Work 2-1
- II.1 Introduction 2-1
- II.2 Background 2-2
- II.3 Current Issues 2-4
  - II.3.1 NASA Research 2-4
  - II.3.2 University Research 2-5
  - II.3.3 Company Research 2-7
- II.4 Conclusion 2-8

III. Device and Task Analysis 3-1
- III.1 Commercial Survey of Companies and Device Analysis 3-1
  - III.1.1 Survey Results 3-2
  - III.1.2 Engineering Category 3-2
  - III.1.3 User Category 3-6
- III.2 Task Analysis 3-10
  - III.2.1 Task Requirements 3-10
  - III.2.3 Description of Actions 3-12
- III.3 Survey Conclusions 3-20

IV. Combining of Device and Task Tables 4-1
- IV.1 Introduction 4-1
- IV.2 Combining Issues 4-1
- IV.3 Mathematical Procedure 4-2
- IV.4 Implementation 4-9
- IV.5 Shared Control 4-12
- IV.6 Conclusion 4-13

V. Conclusions and Future Work 5-1
V.1 Contributions 5-1
V.2 Research Conclusions 5-2
V.3 Follow-on Projects 5-3
V.4 Conclusion 5-5

Bibliography BIB-1

APPENDIX A. A-1

APPENDIX B. B-1

APPENDIX C C-1
C.1 Company Summaries C-1
C.1.1 Joysticks, Mice, Trackballs C-1
C.1.2 Alternative Input Devices C-10
C.1.3 Force Feedback Devices C-18
C.2 Survey Conclusions C-20

APPENDIX D D-1

APPENDIX E. E-1
E.1 Introduction to CHIMERA E-1
E.1.1 Basics of CHIMERA E-1
E.1.2 Starting From Scratch E-2
E.1.3 Modules Developed E-17
E.2 Conclusion E-19

APPENDIX F. F-1
F.1 Hardware Used for Research F-1
F.2 Conclusion F-3

List of Figures

Figure | Page

F.1 Joint Error Plot F-1

List of Tables

Table Page

3.1 DOF Resolution 3-4

3.2 Device Cost Rating 3-7

3.3 Tool Length and Travel Measurement 3-8

3.4 Task Analysis 3-18

4.1 Device to Action Computation Example 4-4

4.1 Device to Action Relationships 4-4

A.1 Engineer Analysis A-2

B.1 User Analysis B-2

D.1 Company Addresses D-1

D.2 Addresses of Companies that Manufacture Alternate Input Devices D-3

ABSTRACT

The purpose of this study is to determine an input device for the Air Force's generic telerobotics architecture for large aircraft maintenance and repair. One area of concern is the human to machine interface, more specifically, which manual controller should be used for the specified tasks in this architecture. I mailed a survey to 68 companies in order to compile a list of possible input devices that the telerobotics architecture could use. 32 companies responded which gave me enough data to generate a list that described the physical traits of the input devices. I then divided the required tasks into actions and analyzed them to generate a list of traits required by an input device. Both the task analysis and device listings were combined mathematically to form a performance table which revealed the possible devices that could perform each individual action.

To aid in development of the Air Force's generic telerobotics architecture, I integrated four input devices into a VME based operating system called CHIMERA. These four devices represent the four different sensor types that are currently available in today's market. The first device is a mouse which relays position changes of the mouse to the computer. The second device is a joystick that can be used in two different ways. The joystick can measure position data of the hand position or it can measure the displacement of the hand from the center of the total movement. The third device is a six degree-of-freedom (DOF) spaceball that measures the amount of force for position data and rotational data. The spaceball allows the user the ability to input six DOFs to the computer simultaneously; however, I discovered that only three DOFs were preferred at a time.

SURVEY AND IMPLEMENTATION OF COMMERCIAL MANUAL CONTROLLERS FOR A GENERIC TELEROBOTICS ARCHITECTURE

I. Introduction

Stripping and painting the exterior of large aircraft is a hazardous task. It is advantageous to remove the person from such tasks and place him or her in a clean non-hazardous environment, by using a robot to perform in such hazardous environments. Since a fully autonomous system is not feasible with current technology in an unstructured environment, a telerobotic system is a plausible solution. A telerobotic system is a system that extends the manipulating capability of a human over some distance. A common use for a telerobotic system is to manipulate nuclear material which is extremely hazardous to a human. One aspect of a telerobotic system is the human-to-machine interface. This thesis has analyzed manual controllers currently available to aid in designing a workable telerobotic system for large aircraft repair and has developed a specification process to select a manual controller for a particular task.

I.1 Motivation

The Air Force is currently developing a generic telerobotics control architecture for large aircraft maintenance. Technologies developed under this architecture will aid telerobotics systems such as space, under-sea, and hazardous area systems to react to nuclear accidents and chemical warfare. Extensive research into telerobotics has been conducted at National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL). JPL accomplished a study for the Air Force which discussed possible control architecture for a telerobotic aircraft repair system (1). The study, titled *A Generic Telerobotics Architecture for C-5 Industrial Processes,* explains the requirements

for the Air Force's telerobotics architecture to handle the different tasks, technologies needed to meet the requirements, and some possible configurations. The study identifies several areas that require more research before the Air Force's architecture is implemented. One such area is the user to machine interface (manual controller) which is the subject of this effort. The study lists capabilities that a generic telerobotics large-aircraft repair system should contain for the system to handle all possible tasks. The manual controller must also follow those same requirements.

The purpose for this research is to define the manual controller requirements and determine if a controller that meets the requirements is available. The manual controller must also be integrated into the generic telerobotics system for large-aircraft repair. Thus, the final stage of this research will be to integrate a manual controller into the current Air Force Robotics Control Architecture.

Since the Air Force is currently down-sizing and cutting back its infrastructure, the development of a manual controller for a specific situation is expensive and may not be the best option. Another option is to look for a less expensive commercial manual controller that is already available. The proposed methodology must locate possible manual controllers which are available in the commercial market. Before choosing a manual controller from the commercial market, the term *possible manual controller* must be defined. This research will define a set of specifications from the set of possible tasks. The specifications can then be used to judge the manual controllers.

I.2 Scope

A set of standards was developed from the tasks that the generic architecture will accomplish. If a manual controller cannot meet these standards, it is eliminated. To develop the set of standards, the processes of the generic telerobotics application to an

aircraft maintenance and re-manufacturing system was analyzed. A set of system requirements and device specifications was developed for each task. The device capabilities and the system specifications were merged to determine which devices should be acquired for the system.

After determining the tasks required for the system and a manual controller chosen for the generic architecture, I interfaced the manual controller into the CHIMERA operating system. Carnegie Mellon University (CMU) specifically designed CHIMERA to be a real time operating system that was for a robotics control application and is operated on a VME bus hardware system (see Appendix E). CHIMERA's ability to handle real time control makes it a possible operating system for the Air Force's generic telerobotics aircraft repair system. There are several different types of manual controllers now available for use. Therefore, several manual controllers were interfaced into CHIMERA.

I integrated four devices into the Air Force Institute of Technology (AFIT) Robotics Automation and Applications Group (RAAG) Lab. These devices are a Thrustmaster joystick, a Logitech mouse with three buttons, a Schilling controller, and a Dimension 6 spaceball. AFIT's hardware used had a major malfunction that I located and corrected (see Appendix E). I then designed software and code to integrate each device into the AFIT system. The thesis was successful in completing all work required to integrate input devices to CHIMERA. The code generated will be delivered to the Robotics and Automation Center of Excellence (RACE) and Armstrong Labs (AL).

I.3 Overview

Several manual controllers was analyzed to determine the device best suited for the telerobotic application to an aircraft maintenance and re-manufacturing system. This research is divided into five chapters.

Chapter 2 consists of a literature review of available devices found in research and of surveys done in the past. It discusses the current research at several universities and companies around the world. It also discusses surveys that was accomplished in the past for many different reasons.

Chapter 3 summarizes the results from the surveys sent to appropriate companies. The acquired data was carefully screened and analyzed to develop a set of capabilities for each device. A description and results of the analysis are discussed in Chapter 3. Some areas that had to be addressed were as follows: the ease of user understanding, ease of reconfiguration by different users, and the cost of the manual controller. Chapter 3 also discusses the tasks of the generic telerobotics application for a large-aircraft maintenance and re-manufacturing system. The tasks that the Air Force's generic architecture must perform was divided into actions. For instance, to paint an aircraft the manipulator must properly attach or grip the end effector, a paint gun in this case, which can be labeled as an action. A set of task specific requirements was developed for each action that was required to accomplish each task.

In Chapter 4, the device and task analysis is combined to determine what the possible devices are for each action. Issues concerned with properly comparing the different devices are discussed and the analysis results are combined.

Chapter 5 discusses the results of this research and possible follow-on projects. The results of this thesis are that for each task there are several actions and those actions each have an input device that is optimal for that action. The next obvious step to this research is to construct a task simulation and test input devices against a task and compare the results of the test with this research.

Appendices A and B contain the device tables. Chapter 4 uses the tables to generate the device to action relationship table. Appendix C contains a table of companies that manufacture input devices and their addresses. Appendix D summarizes each company. Appendix E gives a small tutorial about CHIMERA and how to implement a CHIMERA module.

I.4 Equipment Required

Appendix F discusses the devices that I integrated into the Air Force Institute of Technology (AFIT) RAAG computer facilities. It discusses the hardware needed to connect the devices to the CHIMERA operating system. Appendix G discusses the software to properly integrate the device into the CHIMERA operating system. A demo was developed to show the manual controllers in action. The recommended manual controllers as well as the code used to integrate the device into CHIMERA will be delivered to AL and RACE for use in the generic telerobotics architecture for an aircraft maintenance and re-manufacturing system.

The equipment used during this research was the VME bus controller and PUMA 560 robot running the CHIMERA operating system which is available in the AFIT Robotics and Automation Applications Group (RAAG) Lab B. After correcting a problem with the PUMA controller, four manual controllers were interfaced into the system to ensure the specifications were met. A demo of shared control will also be discussed. A force sensor was used to control action in one direction and the manual controller was used to control action in the orthogonal directions to demonstrate the PUMA writing on a white board using a marker. The demo simulates the PUMA painting a large aircraft using proximity sensors and stripping paint using a force sensor.

II. Previous Work

II.1 Introduction

The Air Force is currently developing a generic telerobotics architecture for aircraft maintenance. The technology developed for this system will aid any telerobotics system developed in the future. Some examples of this aid are telerobotics in space, under sea, and hazardous areas such as nuclear accidents and warfare. Extensive research in this area has been conducted at National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL). One particular study that deals with this architecture is titled *A Generic Telerobotics Architecture for C-5 Industrial Processes* (1). The study discusses the requirements that the Air Force's system must handle, technologies needed to meet the requirements, and some possible configurations. The requirements are derived directly from the tasks the system must perform. The study stated several areas that required more research before the Air Force's system could be realized. One such area is the user to machine interface (manual controller). The study lists several requirements that a generic telerobotics aircraft repair system should have to ensure full compliance with the desired tasks. Tasks that the study described were:

1) painting of the aircraft outer skin.
2) paint stripping of the aircraft outer skin.
3) aircraft outer skin polishing and cleaning.
4) aircraft fuel tank desealing and resealing.

The manual controller must have the ability to translate human movements into realizable robot manipulation to accomplish all possible tasks. The purpose of this research is to develop a specification for manual controllers for a particular task.

II.2 Background

A manual controller is a device that interfaces a human operator to a machine of some sort. The manual controller can be divided into three main parts. The first part is the grip. The grip is the mechanical part of the device that comes into contact with the human hand. According to Bejczy (4), there are 14 types of grips (industry standard, accordion, full-length trigger, finger trigger, grip ball, bike brake, pocket knife, pressure knob, T-bar, contoured, glove, brass knuckles, door handle, and aircraft gun). In some cases, the classes were divided by the placement of the button or trigger. The full-length trigger, finger trigger, bike brake, and pocket knife are classified as hand grip in this paper. There should be at least two more grips that are not covered by Bejczy. They are finger grip which is a small lever attached to the sensor section of the device. The lever is so small that only the fingers can operate it. The second is roller ball. A roller ball is a grip style that is used in a typical trackball. The ball is recessed into the device and only the top of the ball can be actuated by the human.

The second part of the manual controller is the sensor section. This part contains the mechanical motion to electrical conversion. The conversion is accomplished by several methods. Some of the methods are potentiometers, optical sensors, force sensors, toggle switches, or magnetic sensors.

The third part of most manual controllers is the electrical to computer interface. Not all devices have this third part since there are tasks that don't require a computer-ready input. With the increase in computer speed and the ability for the computer to handle real time 3D situations, a fast three degree-of-freedom (DOF) input device is desirable. The standard keyboard to computer interface just is not good enough. Thus, the need for a good analog signal to computer interface is becoming more prevalent.

Jacob, Sibert, McFarlane, and Mullen (11) demonstrated that the device must be matched to the task. They showed that a dimensionally separable task can be performed as well or better with a device that is also dimensionally separable. A dimensionally separable task is defined as a task that requires control in more than one DOF and one axis is independent from one or more other DOFs. A good example of a two-dimension separable task is driving a car. The car is controlled in a two dimensional space and the forward/reverse is independent of the steering wheel control. On the other hand, a mouse is also controlled in two dimensions; however, the two dimensions are independent. The conclusion made by the authors can be interpreted as the all encompassing (six DOF, force feedback, etc.) device may not be the best device for all tasks. It is recommended that each task or set of actions be analyzed to select the device that will handle that task the best. The paper did not look at the entire device space. The authors only looked at the device in the task space. To properly optimize the device selection process, all variables should be analyzed such as power required, size, weight, and most importantly cost. These factors may not seem to be a concern in the research area; however, if that device is going inside the space shuttle, then weight and size are crucial.

Another paper that attempted to model the input device was written by Mackenzie and Buxton (17). The authors analyzed the input devices from a human factors point of view. They examined the performance of an input device by creating a computer tracking test. For certain tasks, this model will work very well. However, for other tasks, like three DOF tasks, this test may not perform that well. There are few three DOF input devices that use the same method for all three DOFs. A good example is a joystick that supplies a third DOF by allowing the joystick an up and down motion. The dexterity and the range is different for the three different DOFs. Also, the test did not take into account all possible variables such as size and weight.

II.3 Current Issues

In this section of the literature review, I examined the latest developments in manual controller design and implementation. The literature review will be limited to technical data only and will not discuss commercial brochures. This section is separated into three parts. The first part will discuss the work accomplished by NASA. The second part will discuss current or past research at some universities. The third part contains research carried out by companies that have written technical documents on manual controllers.

II.3.1 NASA Research. NASA has a long history of research in developing manual controllers (5:Ch 1, 19, 20, 21, 22, 23, 18, 10, 8). NASA's research began in the early 1960's with the development of a mechanical teleoperator which was designed to operate in a hazardous nuclear environment. The research changed to development of manual controllers for satellite and deep-space operations in the late 1960's with the introduction of the Maneuvering Work Platform (MWP). Like most early teleoperated systems, the MWP had a specially designed manual controller that was big, bulgy, and difficult to use. More recently NASA, like the Air Force, can no longer afford to design expensive manual controllers for the vast array of applications. Thus, NASA has accomplished studies to look at commercially available manual controllers to operate their complicated systems.

NASA Langley Research Center performed several tests of rate control input devices to try to understand their performance capabilities (28). The three manual controllers studied were a Honeywell six DOF manual controller, a Kraft six DOF force feedback manual controller, and a manual controller constructed by using two commercially available joysticks with three DOFs each. The study concluded that for certain tasks the cheapest manual controller (two joystick type manual controllers) operated as well as the more expensive manual controllers. However, the study also stated that for a task that

required contact with the environment, the Kraft force feedback manual controller outperformed the other manual controllers.

NASA Langley Research Center also performed a study that established a set of guidelines to properly test a manual controller (8). The study discussed several variables that must be accounted for when performing a manual controller experiment. Some of the variables discussed were indirect and direct viewing of an experimental task. A direct view method is the human subject has the privilege of viewing the task directly while an indirect view method contains at least one camera and the human subject must perform the task by using the camera view only. Another variable is the task being performed. Each task will be different and will have its own requirements that must be accomplished to complete the task. An additional variable is the human subject. Since every human is different, a strict statistical analysis should be accomplished to remove the human variable out of the experiment. The study finished by performing a bolt threading experiment to show a way of testing a manual controller. The test measured time of task completion, number of errors, rate of control motions, and subjective workload. The number of errors was tallied as the number of accidental contact, loss of control, misalignment of bolt to bolt hole, and torque overload. The subjective work load was a survey question that was responded to by the user.

II.3.2 University Research. Most universities studied certain implementations of a type of manual controller or designed a part of a manual controller to test it for use in a complete manual controller. One such study was performed at Harvard University (9). The article from Harvard University discusses the use of a tactile sensor as a manual controller. A manual controller of this kind would be ideal for certain tasks such as rotating a screw, where sensitive force feedback is required to align the screw and to properly rotate the screw. This manual controller had several problems with the human impedance. According to the article, the human impedance varies not only from human to

human but also from task to task. Thus, some impedance matching must be performed in real time during task performance which is not an easy task for current computer technology.

Kazerooni has done a considerable amount of work in the area of human-machine interaction (12, 13, 14) at the University of California. He has constructed several large devices that the human can climb into or fit into, which enhances the humans strength. These systems are by no means graceful enough for application yet, but are a good step in the right direction.

The University of Texas at Austin has constructed a six DOF force reflecting joystick (15). The device uses nine strings to relay position data to the manipulator and force information to the user. The joystick is rather limited in its workspace because of all the strings that are attached to the grip and the enclosure that surrounds the device on all sides but one.

An exotic manual controller was studied at Rice University (6). It is based on new sensors that will measure the amount of human muscle inflection. This type of manual controller would be ideal for an anthopromorphic teleoperated system. The human would simply attach electrodes to his/her body and the slightest motion would be registered by the sensors. Current technology requires donning a special apparatus that can be intimidating to a human operator and can hinder motion.

University of Florida looked at using a robot to control another robot (26). A PUMA 600 was used as a joystick to control another robot. Currently, there are input devices that match the PUMA manipulator (see Schilling summary in Appendix D).

II.3.3 Company Research. H. N. Jacobus, who has started her own company called Cybernet, has classified manual controllers into seven categories (24:Ch 15). The first

category is switches and potentiometers. A good example of this category is a push button. The second category is joysticks, and the third is replica controllers. The third category contains controllers that kinematically match the slave or the controlled robot. The fourth category is master-slave controllers. A master-slave controller is a sub-category of replica controllers that have the same geometry and kinematics of the slave. The fifth category is anthropomorphic controllers. An anthropomorphic controller is designed to fit a human. This type of design tries to use the human's natural reaction to control the slave. The sixth category is nongeometric analogic controllers. This type of controller is like replica controllers in that they have a direct mapping to the slave kinematically, but they don't match geometrically. The seventh and last category is universal controllers. A universal controller relies on some sort of computer to map the controllers motion to the slave. Jacobus has constructed the Per-Force hand controller which is a six DOF force feedback controller. Cybernet was kind enough to demonstrate the Per-Force controller here at AFIT.

Another company that has accomplished some work in force feedback hand controllers is SensAble Devices. Massie, who has started SensAble devices, has constructed the PHANToM haptic interface. Massie is moving forward with his PHANToM to a multiple finger role.

According to a Schilling brochure, Schilling is a company that manufactures a force feedback manual controller. The manual controller is a seven DOF device that reads positions of the grip and delivers forces of the manipulator to the user.

II.4 Conclusion.

Although there are many controllers specially designed for certain systems, there is a recent trend of trying to find a cheaper manual controller that will perform as well as the

expensive ones. This review highlighted several such controllers that could possibly be used in the Air Force's generic telerobotics architecture for C-5 industrial processes and a test that could be used to determine the optimal manual controller. Since recent studies indicate that manual controllers do not have to be specifically designed to meet the requirements of a particular system, then the commercially available manual controllers should be surveyed to find the optimal manual controller for a particular telerobotics system.

III. Device and Task Analysis

This chapter will discuss the survey of all commercially available manual controllers. It will begin with a discussion of the survey responses and how the information from the responses was tabulated. This chapter will also discuss the tasks involved in the generic architecture. The tabulated survey information and the task analysis will be combined to determine the optimal manual controller in Chapter 4.

III.1 Commercial Survey of Companies and Device Analysis

This section will massage the survey information obtained from many different companies. The information will be tabulated to allow easier comparison of each device to a particular task. In most cases, the tabulated information will be multiplied and summed with the task analysis information. In some cases, the particular item will be an elimination item. An elimination item is a characteristic that eliminates a device from further consideration. If a serial interface is the only possible type of communication interface that a system could handle, then any device using any other type of communication interface would be eliminated. Further explanation of the elimination cell will be explained in the task analysis section. Once all devices have been eliminated, the task analysis tables will be multiplied with the device analysis tables and the device with the lowest number will be considered the best device. The resultant matrix will contain device results for each task; thus, a device that is the best for one task may not be the best for another task.

III.1.1 Survey Results. Early in this research letters requesting any information on any possible input devices were sent to over 60 companies. The letter requested that the

company send information on any devices that could possibly be used in a telerobotics applications or in any human to machine interface system. Several companies responded by letter and/or brochure. The information from the responses was tabulated to give a grading of the input devices.

The information was divided into two main categories or point of views. The first category is how a device designer would judge the devices. This category would look at issues or items such as electrical interface, communications type, number of degrees-of-freedom (DOFs), resolution of each DOF, etc.. The second category is concerned with how a user would judge the device. The user is defined as the person using the device. The issues or items observed in this category are size of the device, dexterity, type of grip etc.. Each category will be explained in detail.

III.1.2 Engineer Category. The engineer category is concerned with issues that an engineer would maximize to make the optimal device. Table A.1 in Appendix A tabulates six items of each input device the issues, called items, in the first row and the results of each input device to that issue.

The first item in the table is power. The less the power the better the device. A one in the power item of a device represents no external power supply required while a ten represents a large power supply such as 1 mA @ $\pm$15 volts. A five represents a five volt external power supply is required to operate the input device.

The second item is communication interface type. This item is an elimination cell which will either allow the device to be tested or will eliminate the device from the test. The only possible types of communication interface for this survey are serial, parallel, analog, switches and/or other. Some devices may have two or more different types of communication interfaces, while others may have none which would require additional

hardware to interface to the system. An analog output device will require additional circuitry to interface to a computer, such as an A/D board or an analog to serial interface card. This situation may be misleading to the outcome of "optimal device" because the cost to interface a device to the computer will be included in some devices but not others. Thus, a cost adjustment must be made to even the score. Based on an average of interfaces and A/D chips, the price should be increased by $100 and the weight increased by one pound. These increases are only estimates and are not the cost and weight of a constructed product. The switch interface type is used to reveal a toggle switch joystick. A toggle switch joystick is a joystick that operates only by opening and closing switches. There are only on or off signals from a toggle switch joystick.

The next item in the table is DOFs which is the amount of DOFs the device can create. This item does not include the number of buttons, even though buttons can be used to generate a DOF.

The resolution per DOF item is defined as the resolution of the device per DOF to the system. Some devices use open-close switches to develop a DOF signal (toggle switch joystick); thus, the resolution for that DOF is one bit which corresponds to a ten for this item. Other devices convert sensor output to a hex value that can be read by a computer. See Table 3.1 to determine the resolution per DOF.

Table 3.1 DOF Resolution

Range of hex Values	Resolution per DOF
1	10
2-16	9
17-64	8
65-144	7
145-256	6
257-400	5
401-576	4
577-1024	3
1025 or Greater	2
Infinity	1

Since the computer must read the input device signal and cannot read an analog signal directly, the value of infinity was included only to grade the device. In some cases, the actual resolution would be determined by the type and resolution of the interface device. For example, if a device supplies a voltage then it could have an infinite resolution. The device would then be interfaced to the systems computer. For this example, assume an A/D is used to interface the device to the computer. The actual resolution would be the range of hex values the A/D can supply to the computer. Some devices only output a one if a certain distance is traveled. This type of device, mouse for instance, is normally rated in dots-per-inch (dpi) or pulses per revolution (ppr).

The next item in the engineer table is the number of buttons. A button on a device can be very useful. They could be used as an emergency shutdown, mode change, or a simple check to ensure the user has a grasp on the grip. Buttons could also be used to generate an additional DOF. It is recommended that two buttons be used to generate a DOF, if required, one button for increase and the other for decrease.

The last item on the engineer table is the type of device. The possible values of this item are joystick, mouse, trackball, or other. Other describes an input device that does not fit the other three device types. The Immersion interface device (see Appendix C) is a good example of this device type.

There were three other items that were not included in Table A.1 which are communication speed, signal stability, and temperature tolerance. The items were not included for two reasons. First, there was little information available for these items. A second correspondence was required for all but three devices. Secondly, if the information was available, it was well within the human limits of operating an input device. Even though the next three items were not covered in Table A.1, the description and analysis for these items are still included.

The first item not included is communication speed. The communication speed is defined as the total time it takes the system to request data to the time it takes the system to receive the data. The communication speed does not take into account the amount of time it takes a computer to perform a read or a write, because in all devices the processor is required to access the communication device. For a serial type device, the total time would be the amount of time it takes to send the data plus the latency time plus the time it takes to receive the data. Latency time is the time it takes the movement of the device to become available to the electronics. In equation form, the total time is:

$$\text{Total Time } = 2(\#\text{bits} / \text{baud rate}) + \text{latency time} \qquad (1)$$

Since only a read is required to get the data from the parallel or A/D, the total time is the latency time of the device.

Another possible item is the signal stability. This item is an objective item that is concerned with signal to noise ratio, signal accuracy, and linearity of the signal. The better the signal stability the lower the number. A one corresponds to a very good signal-stable device and a ten would correspond to an unstable signal.

Another item that may be of interest in some cases is the temperature tolerance of the device. In most cases, the telerobotics system will maintain a controlled environment for the human which is less able to handle a large temperature range as compared to a physical device. A one would represent the widest temperature range of -20 degrees C to 120 degrees C. A ten would represent the smallest temperature range of 20 degrees C to 50 degrees C.

III.1.3 User Category. The other category that the device information is divided into is user analysis. The user analysis is concerned with how the user judges the device. An ideal device will feel transparent to the user while a bad device will cause the user a tremendous amount of grief. Though most items in this category are subjective, they are most likely more important items then in the engineer category. In most cases, the items in the engineer category can be adjusted (engineering trade-off) to the requirements of the system. If the device does not feel right to the user, then the task will not be performed in the best manner. The results are tabulated in Table B.1 in Appendix B.

The first item in this category is the cost of the device. The values in this item will go from one to ten where one is the lowest price and ten is the highest price. Since the range of prices is large, this item will be divided into 20 parts instead of ten like the other items. The divisions are shown in Table 3.2.

Table 3.2 Device Cost Rating

Price ($)	Item Number
0-100.00	1
100.00-200.00	1.5
200.00-300.00	2
300.00-400.00	2.5
400.00-500.00	3
500.00-600.00	3.5
600.00-700.00	4
700.00-800.00	4.5
800.00-900.00	5
900.00-1000.00	5.5
1000.00-2000.00	6
2000.00-3000.00	6.5
3000.00-4000.00	7
4000.00-5000.00	7.5
5000.00-6000.00	8
6000.00-7000.00	8.5
7000.00-8000.00	9
8000.00-9000.00	9.0
9000.00-10000.00	9.5
10,000.00 or greater	10

The next item in the user table is reliability. This item is difficult to obtain from a company catalogue or brochure. If reliability is important for the task; then, further information should be obtained from the company or from someone who has used the device for an extended period of time. There are certain characteristics that will help in determining reliability. A device is only as good as its weakest part. For instance, if the joystick sensors are potentiometers, then the reliability would not fare as well as a joystick with inductive sensors, on average. In some cases, reliability information can be obtained on the device from a catalogue or brochure. Such information might be described as cycle limit. Other brochures may describe the reliability in terms of Mean Time Between Failure (MTBF). For the purpose of this research, the reliability will be a number between one and ten. A one represents good reliability and a ten represents poor reliability.

The third item in the User table is dexterity. Dexterity describes the physical readiness and grace of a device which can be very subjective. A dexterity measure can be determined by observing the physical characteristics of a device. The size of the tool is a good measure. If the tool size is small, then the dexterity measure is small. Likewise, if the tool is large, then the dexterity measure is large. The tool is the object that is in direct contact with the human while the device is being used. The amount of travel the tool has is another good measure. In the case of a joystick, if the tool travels ± 20 degrees then the dexterity measure would not be as good for a tool that travels ± 60 degrees. In the case of a trackball style device, the amount of travel is infinite. These two physical characteristics should be combined in a way that will not cause one characteristic to out weigh the other. The following guidelines are given to aid in determining the dexterity measure. The size of the tool and the travel of the tool is broken into five sizes as shown in Table 3.3.

Table 3.3 Tool Length and Travel Measurements

Tool Length	Measurement	Tool Travel (Degrees)
under 1 cm	5	under 20
1 to 4 cm	4	20 to 60
4 to 8 cm	3	60 to 100
8 to 12 cm	2	100 to 180
longer then 12 cm	1	over 180

The two measures should be added to make the total dexterity measure. If a device has a tool that is five cm long, a measure of three, and travels ± 35 degrees, a measure of three, then the total dexterity measure would be six. A trackball tool would have a tool length of zero and a tool travel of infinite thus the total dexterity for a trackball is six. The same rules that pertain to the trackball can be used for the mouse. The mouse tool length

is zero, a measure of five and the mouse travel is infinite which corresponds to a measure of one. The total measure for a mouse is six. The dexterity measure for a trackball and a mouse are the same which, depending on application and/or personal preference, may not be true. There are arguments for each device why it should be better than the other. For this research, both devices are considered to have the same dexterity measure.

The next item in the user table is the grip. As described in Chapter 2, there are several grip styles. This elimination item will identify what kind of grip the device has and the task table will allow those desired grips to be tested.

The next two items in the user table are size and weight. For most telerobotics tasks, size and weight are not an issue; however, if the device must be carried into space or placed in a small room, then size and weight become an issue. For this research, the size and weight was distributed linearly into ten different measures. The smallest and lightest device being a one and the largest and heaviest being a ten. To determine the size, the entire workspace was considered. For instance, a joystick has a base and a tool. The size of a joystick is the total height of the joystick, height of the tool plus the height of the base, times the square of the maximum of either the tool travel width distance or the base width.

The next item, which is an elimination item, is self-centering. The self-centering item is a yes or no item that is used to determine if additional safety precautions must be used to interface the device. The basic question this item answers is will the device fall or move if the user removes the hand or lets go of the device. This item also tells if the device is a position input or a velocity input. A self-centering device is usually thought of as a velocity control device and a non self-centering device is a displacement or position input device. If the device centers itself or returns to a standard position, the item is yes.

The previous sections described the devices and how they were tabulated into the engineer table and user table in Appendix A and Appendix B, respectfully. The next section will discuss the tasks required by the Air Force generic telerobotic architecture. The task will be tabulated into a form that will match the device tables (Table A.1 and Table B.1). The data from this chapter will then be combined (see Chapter 4).

III.2 Task Analysis

This section will discuss the tasks that must be performed by the Air Force's generic telerobotics project. It will discuss the process level tasks by breaking them into specific actions. The specific actions will be analyzed to develop a set of desired input device requirements. The desired requirements will then be described in a manner that matches the device tables in Appendices A and B.

The tasks for the Air Force's generic telerobotic architecture are defined in a report written by the Jet Propulsion Laboratory (JPL)(1). The document lists six process level tasks that must be accomplished by the system and must use an input device. Since the telerobotic system is generic, the system may perform a different set of tasks. The generic telerobotic system could possibly be used in an entirely different set of tasks in the future. For this research, the input devices will be graded against some of the tasks specified in the JPL documents.

III.2.1 Task Requirements. The first process level task listed in the JPL document is painting of the C-5A/B exterior in a dedicated hanger facility. The requirements that apply to the input device for this task are as follows:

1. Perform 90% of paint application with little or no repainting.
2. Apply primer between 0.5-1.5 mils ±0.5 mils.

3. Apply paint between 2.0-3.0 mils ±0.5 mils.
4. Perform flare offs and other painting patterns to smoothly merge adjacent painting areas regardless of drying.
5. Provide separation/standoff distance accuracy of ±1.0 inch, repeatability of 98% in supervised-autonomous control modes.
6. Provide end-effector tool angle of incidence normal to the surface being painted, with an accuracy of ±5 degrees in shared, supervised-autonomous and autonomous control modes.

The second task is painting of removed parts in a robotic workcell. The requirements for this task are the same as the previous task with the exception of the flare off requirements. It is assumed that the entire part will be painted thus no flare offs are required. Another difference from the previous task is the mobile/crane is not required to place the manipulator into place. It is assumed the part is in place and the manipulator has full access to that part.

The third task is paint stripping of a C-5A/B in a dedicated hanger. The specific requirements for this task are as follows:

1. Maintain a standoff distance of 18 to 24 inches with a ±2 inch tolerance and accuracy of ±1 inch with a repeatability of 98%.
2. Direct and control particulate (e.g. plastic material bead, water, glass bead, CO_2) blast pressure and material flow rate.
3. Provide selectable end-of-arm tooling angle of incidence to the aircraft surface normal of 0-45 degrees, with an accuracy of ± 5 degrees in shared, supervised autonomous and autonomous control modes.

The fourth task is surface finishing in the form of removing material from patches and polishing metal to a high gloss finish in a robotic workcell. This task is unique among the other tasks because surface contact is required. The other tasks require a standoff distance. The specific tasks are as follows:

1. Provide surface contour following for unmodelled parts with a selectable force application range in shared and supervised-autonomous control modes.
2. Accommodate the removal of 10-20 mils (not exceeding the paint alodine boundary) of material, with and application force boundary no greater than that required to remove the paint.

The fifth task is surface cleaning of removed parts in a robotic workcell through application of a bicarbonate of soda particulate stream. The specific requirements for this task are as follows:

1. Provide separation distance accuracy of ± 1 inch repeatability of 98% in shared and supervised-autonomous control modes.
2. Provide selectable end-of arm tooling angle of incidence to the part surface normal of 0-45 degrees, with an accuracy of ± 5 degrees in shared and supervised-autonomous control modes.

III.2.2 Description of Actions. The tasks specified above can be divided into two different groups of tasks. The first group is a set of tasks that is performed in a dedicated hanger. The second group is a group of tasks that is performed in a robotic workcell. To aid in developing a set of specific requirements for an input device, a set of actions must be defined to accomplish a group of tasks. The actions required for group one tasks (tasks in a dedicated hanger) and an explanation of the requirements are listed below.

1. Grasp or attach the proper tool to the manipulator. Since all objects in this action can be structured, the input required by the human is a simple *get tool* command. An easy way to implement such a command would be to have a list of icons or a menu that the human can choose. For example, a menu with such options as attach paint gun, attach grinder, attach water sprayer, attach bicarbonate stripper, and attach polish tool could be used so the user could drag the pointer to the appropriate item on the menu and the selected tool would be grasped or attached to the manipulator. Since human to screen interface is required, at least a two DOF input device is needed. The device must also have one or more buttons to activate the icon and/or menu item. There are no limiting factors on power, dexterity, grip, size, weight, self-centering, resolution, and type of device other then user preference. The limiting factors will be comm. type, cost, reliability, DOFs, and number of buttons. To interface the device to CHIMERA, the comm. type must be serial, parallel, or A/D which allows most devices to be used. The cost should always be a minimum and the reliability should always be a maximum. The task analysis table (Table 3.4) shows the resulting elimination cells and weights for the desired input device.

Table 3.4 Task Analysis

Action	1	2	3	5	6a	6b
Power	5	5	5	5	5	5
Comm. Interface	Serial	Serial	Serial	Serial	Serial	Serial
Cost	100	100	100	100	10	10
Reliability	10	10	10	10	10	10
Dexterity	1	50	50	1	100	100
Grip	X	X	X	X	X	X
Size	1	1	1	1	1	1
Weight	1	1	1	1	1	1
Self Centering	X	X	X	X	Y	Y
DOF	≥1	≥3	≥3	≥1	≥3	≥2
Resolution per DOF	X	≤6	≤6	X	≤6	≤6
# Buttons	≥1	≥0	≥1	≥1	≥1	≥1

2. Gross movement of manipulator to the task area. The manipulator will be mounted on some sort of movable base. The options are a tele-crane, gantry, or mobile system (1). In any case the manipulator, once loaded with the appropriate tool, must be moved to the aircraft and positioned such that the work can be accomplished. Since this is a rough movement requirement, the accuracy and repeatability will not be a factor when deriving the user input specifications. The problem with this action is the requirement that 90% of the aircraft must be accessible by the manipulator and the aircraft is not easily accessed in all locations. A mobile system is currently being examined by the Air Force for the rough positioning system. More specifically a scissors truck with a manipulator mounted on the top seems to be the choice. If this is the case, then moving the mobile unit or scissors truck will be a difficult task. The scissors truck cannot be moved by a simple movement of a joystick (or any other input device) because the scissors truck does not have complete freedom in both directions of movement. The

scissors truck must be moved forward and backwards with the front wheels turned in the appropriate direction. This may cause several problems when a user is trying to place the scissors truck into a position that is relatively close to the aircraft. There are several ways that this problem can be solved. One possible solution is to place the steering mechanism on all four wheels thus giving the scissors truck more freedom to move in two directions. This solution may be relatively expensive when compared to the price of the scissors truck. Another possible solution is to allow the user to pick a location on a screen and the computer computes a recommended path. This solution may not be viable in all situations. The user should still have ultimate control of the scissors truck. In any case, obstacle avoidance is still an issue that must be solved with additional sensors to prevent damage to the aircraft. The important items are listed in the task analysis table (Table 3.4) which show the elimination items and weights for the corresponding device analysis tables. The limiting factors for this action are comm. type, cost, reliability, dexterity, and DOFs. The dexterity is more important on this task because of the scissors truck maneuverability problem. Since the scissors truck can be moved in two dimensions and lift in the third dimension, a three DOF or higher input device is required.

3. System instructions from operator through interface to system. This specific action will be used if supervisor-autonomous modes of operation is desired. The user will input a set of coordinates, some additional instructions, and the manipulator will perform the task. An example of this can be explained by describing this action while a user is stripping an aircraft using bicarbonate. The user would pick four or more corners on a video screen that represents at least four corners of the area that needs to be stripped. The computer would then compute the best possible trajectory. Keep in mind the trajectory is only two dimensional. The third

dimension is controlled by either a force sensor or proximity sensor which is called shared control. The shared control theory has been implemented at AFIT to determine its applicability to this scenario. Again, the problem with this action is the requirement of completing 90% of the aircraft. The aircraft is not square or in straight lines. The trajectories required may be difficult to compute and accomplish in some areas of the aircraft.

To pick the corners of the workspace, the user must move the robot to those corners. The user will most likely be looking at a TV monitor and moving the manipulator to the desired point. If a force feedback device is not used, then an indication of contact or proximity tolerance must be supplied to the user. The input device must be at least a three DOF device. The user could move the manipulator into location in a two dimensional space then activate some kind of force control or proximity algorithm that would move the manipulator until it contacted the aircraft. In either case, the user must maneuver the manipulator into a point. The point may be an adjacent point to the last section of work area or a point that is close to an obstruction on the aircraft. Thus accuracy must be maintained to prevent aircraft damage. The end effectors used in this group of tasks inherently have some tolerance. For example, the bicarbonate stripper has a spray width of approximately three to four inches. The paint spray has a width of one to four inches depending on the type of sprayer used. Even with these tolerances, the points selected by the user should have an accuracy of 0.25 to 0.1 inches. This accuracy can be easily obtained with any device that is a displacement device. The software could be modified to control the manipulator distance for an associated distance from the input device. An example of this would be to use a mouse to move the manipulator one millimeter. The software could require the mouse be moved several inches

before the manipulator is moved a single millimeter. On the other hand, a single pulse from the mouse could represent several inches to the manipulator. If the above could be adjusted by the user, then perhaps the user could select a more accurate movement when needed and a faster movement when accuracy was not an issue. Thus, some sort of throttle control could be desirable to the user. An accepted method among the computer group of adjusting the speed of the mouse is called ballistic. The more the mouse is moved in one direction the faster the pointer moves.

4. Monitor task using sensor devices and appropriate feedback to user. This action does not require an input device to accomplish. It will be handled by the system and appropriate output devices.
5. Error detection and recovery. This action requires one input from the user. A safety button or switch should be used to accomplish this action. A button on the input device would be a viable option to accomplish this action, yet risky. If the user accidentally pushed the wrong button on the input device, then an unwanted shutdown would occur. It is recommended that a separate, rather large, button be used to accomplish this action. The large button would be easy to locate and activate under an emergency or a fault situation.
6. If required, fine motion of manipulator to accomplish required task. This action is required if for some reason the equipment fails, the system cannot handle one of the many variables, or the user deems it necessary to control the manipulator manually. Since the orientation and position is a required control variable (see task requirements above), a six DOF input device or two, three DOF input devices are needed with all possible sensor data displayed to the user. Such a system would give complete control to the user. However, in some cases the input device would only control two DOFs while the proximity or force sensor controls the others.

Since this action can be divided into two different methods of accomplishment, it will be separated into two actions. The first action (6a) is complete control of the manipulator by the user. The second action (6b) will be shared control between the input sensors and the computer, and the user. As stated earlier, action 6a requires a six DOF input device that is fairly dexterous and easy to use by the user. Action 6b will only require a two DOF input device. Action 6b has been implemented in the AFIT robotics lab to demonstrate shared control while simulating painting an aircraft skin. Appendix F describes the system used to perform action 6b. This action requires a rather dexterous two DOF input device and at least one button to activate the sprayer or stripper. A good example of this action is painting an aircraft. The user can control the paint gun in two dimensions while the computer and input sensor(s) (proximity or force sensor) control the other DOFs.

The second set of actions pertaining to group two requirements in a robotic workcell are a subset of group one actions. The same actions are required with the exception of moving the mobile platform into place. The tool grasping or attachment, system instructions from operator to system, monitoring of the task, error detection and recovery, and finally fine motion of the manipulator by the user are all still required actions.

Table 3.4 displays the appropriate weights and eliminations for each action. Some items described in the device tables are not included in the task analysis table because they were not a concern for the list of actions required to accomplish the tasks. For instance, the item "Type" was not included because it didn't matter what type the device was as long as could perform the necessary action. Some items are the same for all actions which is due to the action requirement and/or the hardware. For instance, the power, and comm.

interface items are limited by the hardware used. If the only means of a system to communicate to an input device is through a serial port, then all actions (second row of Table 3.4) should have serial as the comm. interface.

Some items are called elimination cells because these items describe a hard limit that if not met then the device must not be included in the table. The numbers in the task analysis items indicate a weight associated with that item.

The rows in Table 3.4 represent the weights for a corresponding item in the device tables (see Appendix A and Appendix B). The first row is power. Since our system is maintained on the ground in a controlled environment, the weight for this item is five for all actions. The second row is comm. interface. This elimination item requires the device to have a serial output of some sort. As mentioned earlier, an analog device can be connected with the proper hardware which will add to the cost, weight, and size of the device. The third row is cost. Cost is important to the Air Force so a weight of 100 is in most of the actions. However, the cost is lowered for actions that require high dexterity to make the dexterity weight more valuable. The fourth row is the reliability. The weights for all actions for this item is ten. The ten represents reliability which is an important factor, but not the most important. The fifth row is grip. The grip does not matter in this analysis as long as the actions are able to be completed reasonably and is so indicated by an X in all actions in the fifth row. The sixth row is dexterity. Dexterity is not an issue for the first five actions; thus, the weight is only ten. However, for the last two actions in the sixth row of Table 3.4 dexterity is important which is shown by the weights of 100. The seventh and eighth rows are size and weight, respectfully. Since the input device will be in a controlled environment with unlimited room and unlimited weight constraints, the weights for this item are ones. The ninth row is self-centering. Since position control or velocity control is not an issue, the elimination item is labeled as don't cares for all but the

last two actions. The last two actions contain a yes because if the user were to let go of the input device, then the aircraft may be damaged. The last three rows of Table 3.4 represent minimum or maximum requirements. These last three rows are used as elimination items to pick only the input devices that meet or exceed the requirements.

III.3 Survey Analysis Conclusions

This chapter described the commercially available devices in tabular form. Since no one device is the best device for all tasks, the devices must be correlated to specific tasks. The next chapter combines the device tables (Table A.1 and Table B.1) and task table (Table 3.4) to show comparisons between the input devices.

IV. Combining of Device and Task Tables

IV.1 Introduction

This chapter will take the results of Chapter 3 and combine the information into a table that represents the input devices to task relationships (see Table 4.2). The end result is a set of possible input devices per action. In this chapter, the mathematical procedure is discussed and issues that must be resolved before the tables can be combined. A simple example of comparing one device to a task is also accomplished to aid the users understanding of the mathematical procedure.

IV.2 Combining Issues

Before the tables can be combined, some issues must be resolved. The first issue is how will the analog input devices be interfaced to the system. One possible solution is to use an A/D to digitize the analog signal. A fast reliable A/D board will be relatively expensive compared to some of the input devices evaluated in this report. However, if expense is not a large factor, then the A/D option would be an option. The A/D board would most likely be the fastest and most reliable of the options. Another option is to construct an interface device. The interface device interfaces the input device to any possible input port on a computer. Some possible input ports are serial port (RS-232, RS-422), bus port, parallel port, etc. A common interface is from the input device to a serial port; probably because most computers have a serial port. Like the A/D board, a fast and reliable interface would be relatively expensive and would not be as fast as the A/D board. The reason the interface takes longer time then the A/D board is because the interface device must digitize the signal then supply the serial line with a stream of bits. Where as the A/D board just digitizes it and supplies the data to the data bus of the computer. The actual time, or absolute time, of either interface is not the issue. The issue is the speed of

one verses the other or relative time. For the purpose of this research, if an input device uses the A/D board then the price will be increased by $100.00, the comm. speed will be increased by 0.0005 seconds, and the DOF resolution will have a maximum resolution of 255 which corresponds to a six in the device tables. If the serial interface device is used, then the cost will be increased by $100.00, the comm. speed will be increased by 0.0008 seconds, and the maximum resolution will be six. These numbers are only estimates.

Some input devices use toggle switches as the motion sensor. These sensors contain a binary output that may not be acceptable in all cases. Thus, another issue is should a toggle switch type input device be used and if a toggle switch device is acceptable then how should it be used. If a task only requires a constant velocity control, then a switched input device would work well. On the other hand, if position control or non-constant velocity control is desired then a switched input device is not a desirable input device. The switch works as a binary device that converts position data into on/off signals which does not lend itself to a position control system. The actions described for the telerobotics architecture requires a non-constant velocity control and/or requires position control, thus the input devices that have switches as their sensors will not be included in the resultant table.

IV.3 Mathematical Procedure

The combining of the device tables and the task table is accomplished by a mathematical approach. The device tables (Table A.1 and Table B.1) are summed by concatenating the rows making a m by n matrix where m is the number of items and n is the number of input devices. The complete device table (matrix) is then multiplied by the task table (Table 3.4) which acts as a weighting matrix. The resultant matrix is the input device to action relationship (see Table 4.2). Another way to look at the tables is to look

at an example. The input device model 215 manufactured by P Q Controls has the numbers extracted from Tables A.1, B.1 and 3.4 and listed in Table 4.1.

Table 4.1 Device to Action Computation Example

Item	From Table 3.4	From Tables A.1 and B.1	Adjusted for Serial Interface	Col. 2 X Col. 4
Power	5	5	5	25
Comm. Interface	Serial	Analog	**Serial**	OK
Cost	100	1	**1.5**	150
Reliability	10	2	2	20
Dexterity	50	3	3	150
Grip	X	Hand	Hand	OK
Size	1	5	5	5
Weight	1	3	**4**	4
Self Centering	X	Y	Y	OK
DOF	≥3	3	3	OK
Resolution per DOF	≤6	1	**6**	OK
# Buttons	≥0	2	2	OK
				354

The table above is an example of computing the device to task relationship. The number at the lower right corner matches the number for action two (column 4) and the P Q Controls model 215. If the device tables (Table A.1 and Table B.1) and task table (Table 3.4) were used as matrices, then the solution of all devices and all actions are simply a matrix multiplication problem.

The following table reveals the device to task relationship. The table consists of numbers or an E. The E is used to describe a device that did not meet one or more of the

elimination items. For example, if an input device only had two DOFs and the action required three DOF then that input device would show an E for that action. The numbers represent the device to action rating. The number is obtained by multiplying the task table (Table 3.4) weights to the concatenated device tables (Table A.1 and Table B.1). The absolute value of the number is not as significant as the relationship of a number to that of another device. The higher the number in the table the less likely the input device in question is the optimal device for that particular action. The lower the number the more likely that input device is the optimal input device for that particular action.

Table 4.2 Device to Action Relationships

Company Name	Model	1	2	3	5	6A	6B
Applied Resources Corp.	Digital	E	565	E	E	E	E
	Analog	E	561	E	E	E	E
Appoint	MousePen	185	E	E	185	E	E
	Gulliver	230	E	E	230	E	E
	Thumbelina	184	E	E	184	E	787
Assmann Data Products	Digitus Magic Click	173	E	E	173	E	677
CH Products	200 MK-III	285	E	E	285	E	501
	400 MK-III	285	481	481	285	496	501
	0 Inductive	E	E	E	E	E	E
	1 Inductive	E	E	E	E	E	E
	2 Inductive	E	E	E	E	E	E
	4 Inductive	E	E	E	E	E	E
	DT225	238	E	E	238	E	697
	P150	E	E	E	E	E	E
	P200	E	E	E	E	E	E
CIS Graphic & Bildverarbeitung	DIM6	404	698	698	404	683	683
CTI Electronics Corp.	F1000-N2	E	E	E	E	E	E
	F1000-N5	E	E	E	E	E	E
	F2000-N2	E	E	E	E	E	E
	F2000-N5	E	E	E	E	E	E
	H0000-N5	E	E	E	E	E	E
	H0800	E	511	E	E	E	E
	H3003-N81	E	E	E	E	E	E
	H8000-N81	E	E	E	E	E	E
	H9000-N81	E	E	E	E	E	E
	M1000-N24	257	E	E	257	E	672
	M3000-N24	258	E	E	258	E	672
	M4010-N24	258	E	E	258	E	672
	M8003-N24	258	E	E	258	E	672

Table 4.2 Device to Action Relationships cont'd

Company Name	Model	1	2	3	5	6A	6B
Cyber-tech, Inc.	8900/500	368	E	E	368	E	341
	8600/500	368	E	E	368	E	341
	1800/500	468	E	E	468	E	351
	1200/500	468	E	E	468	E	351
	BRUT-88/500	368	E	E	368	E	341
	300-H/500	368	E	E	368	E	341
Happ Controls, Inc.	56-5500	E	E	E	E	E	E
	56-0100	E	E	E	E	E	E
Hed, Hydro Electronics Devices Corp.	DAL-002	E	E	E	E	E	E
	DAL-006	E	E	E	E	E	E
IZU Products Co.	MACFLY	198	E	E	198	E	504
Kensington	Expert	228	E	E	228	E	588
KeyTronic Corp.	Professional	226	E	E	226	E	586
	Honeywell	186	E	E	186	E	591
Kraft Systems Inc	Mouse	188	E	E	188	E	593
	Trackball	177	E	E	177	E	593
Logitech	Mouse	177	E	E	177	E	582
	Space Control Mouse	172	466	466	172	685	685
	MouseMan	184	E	E	184	E	625
	Trackman	178	E	E	178	E	583
	WingMan	180	E	E	180	E	684
Maurey Instrument Corp.	JSP	E	692	E	E	E	E
	JSE	E	E	E	E	E	E
	HDJ	E	E	E	E	E	E

Table 4.2 Device to Action Relationships cont'd

Company Name	Model	1	2	3	5	6A	6B
Measurement Systems, Inc.	462	E	E	E	E	E	E
	469	E	E	E	E	E	E
	467	E	E	E	E	E	E
	465	E	E	E	E	E	E
	470	E	E	E	E	E	E
	463	E	647	E	E	E	E
	473	E	647	E	E	E	E
	446	305	E	E	305	E	872
	435	E	E	E	E	E	E
	521	E	E	E	E	E	E
	531	E	E	E	E	E	E
	523	E	E	E	E	E	E
	546	308	E	E	308	E	479
	547	308	453	453	308	474	479
	402	309	E	E	309	E	579
	570	E	E	E	E	E	E
	575	E	E	E	E	E	E
	615	E	E	E	E	E	E
	625	289	E	E	289	E	609
	626	E	E	E	E	E	E
	STX	E	E	E	E	E	E
	JTX	312	E	E	312	E	874
	XLT	261	E	E	261	E	880
Merit, J.R. Controls Inc.	NSO	E	E	E	E	E	E
	NS2	E	E	E	E	E	E
	CSO	E	E	E	E	E	E
	CS2	E	E	E	E	E	E
	MO	E	E	E	E	E	E
	M2	E	E	E	E	E	E
	SL-M0	E	E	E	E	E	E
	VDF	E	E	E	E	E	E
Microsoft Corp.	Mouse	180	E	E	180	E	684
	BallPoint	171	E	E	171	E	783
MicroSpeed Inc.	PC-Trac	238	E	E	238	E	697
	Micro-Trac	174	E	E	174	E	678

Table 4.2 Device to Action Relationships cont'd

Company Name	Model	1	2	3	5	6A	6B
Mouse Systems Corp.	NewMouse	177	E	E	177	E	681
	OmniMouse	177	E	E	177	E	681
	PC Mouse III	177	E	E	177	E	681
	PC Mouse	177	E	E	177	E	686
	PC Mouse 3D	327	621	621	327	696	696
	WhiteMouse	228	E	E	228	E	686
	PC Trackball	226	E	E	226	E	685
OEM Controls, Inc.	JS1	E	E	E	E	E	E
	JS2	E	E	E	E	E	E
	JS5	E	E	E	E	E	E
	JS6	E	E	E	E	E	E
	MS2	E	E	E	E	E	E
	MS4	E	E	E	E	E	E
Polar & Pole Inc.	ATB600	275	E	E	275	E	689
	ATB1200	273	E	E	273	E	687
Prohance Technologies	Mouse	236	E	E	236	E	695
	Trackball	186	E	E	186	E	690
PQ Controls Inc	215	207	354	354	207	364	369
	220	207	354	354	207	364	369
Sauer-Sundstrand Electronics Systems	MCH	220	E	E	220	E	577
Suncom	ICONtroller	E	E	E	E	E	E
Alternative Input Devices							
Immersion Human Interface Corp.	PROBE-IC	E	682	E	E	E	E
	PROBE-IX	E	722	E	E	E	E
	PROBE-MD	E	782	E	E	E	E
Schilling Development Inc.	TITAN	1011	1109	1109	1011	354	E
Spectra Symbol Corp.	MEMBRANE	195	E	E	195	E	759

IV.4 Implementation.

The main goal of this research was to select possible input devices or manual controllers that could perform the tasks that the Air Force's generic telerobotics architecture had to accomplish. Most surveys done in the past were only concerned with a particular task or a comparison of one task to another verses a certain number of input devices. This research looked at possible commercial input devices and compared physical characteristics against the specific tasks that the Air Force's generic telerobotics architecture must handle. To aid in implementation of an input device, four input devices were interfaced to AFIT's CHIMERA operating system. This section will give a set of operating procedures for a user to connect and operate the implemented input devices. These operating procedures assume the user understands the UNIX operating system, how to compile code with the CHIMERA compiler, can operate CHIMERA, and has a basic understanding of serial ports and electrical engineering in general. Appendix E gives a full explanation of starting CHIMERA and how to set up the environment to compile CHIMERA code.

To operate the Thrustmaster joystick in joint space, the user must first connect the hardware then compile and run the software. The Thrustmaster has two physical parts, the joystick and the throttle. The joystick connects to the throttle with the attached cable. The user must then connect the throttle to a power source with the power cable. The order of cable connection is important because of the self initialization that the Thrustmaster performs. The throttle serial cable is then connected to the CHIMERA processor card. The AFIT hardware configuration has two processor cards, each having two serial ports located on the front bottom of the card. The appropriate reconfigurable configuration file (in this case *jjoy.rmod*) will determine which processor and port the serial cable should be connected. I configured the code and hardware to use the top port

of the processor labeled *control*. To change the port or the processor the user must change the *rmod* file. The SIO_DEVICE and SIO_PORT variables defined in the *rmod* file control which port and processor are used in the module. To operate on *control* the SIO_DEVICE must be defined as *isio0*. To operate on the processor labeled *crusher* the variable SIO_DEVICE must be defined as *isio1*. Each processor contains two ports define as *0* or *1*. To operate with port *0*, (top port) which is a modem RS232 standard serial port, the variable SIO_PORT must be defined as *0*. To operate on hte bottom port which is a console RS232 standard serial port, the variable SIO_PORT must be defined as *1*. After compiling the module and linking it into the main program, the user simply spawns the module by typing *spawn control jjoy*. The user must then turn the module on by typing *on jjoy* which starts the module and the joystick control. The code starts in a joint one control mode. The user controls the joint by moving the joystick to the right or the left. When the trigger button is pressed the next joint is controlled. If joint six is the active joint, then pressing the trigger causes joint one control which repeats the cycle.

The Thrustmaster can also control the PUMA manipulator in Cartesian space. The user must connect the same hardware with the same configuration as described in the previous paragraph. The difference is the module *invkin* must be started prior to starting the *cjoy* module. The *invkin* module will perform inverse kinematics to direct the PUMA manipulator's reference positions. Once the user starts the *cjoy* and *invkin* modules the PUMA manipulator will move in Cartesian space. The joystick will control the Y and Z coordinate frames first. If the trigger button is pressed then the controllable axis become X and Z. If the trigger button is pressed again the controllable coordinates become X and Y. Pressing the trigger button once more returns the controllability back to Y and Z coordinates and the process is repeated.

Like the Thrustmaster joystick the DIM 6 spaceball can control the PUMA in both the joint space and Cartesian space. The hardware must be configured as explained earlier. The user must ensure the spaceball is configured for proper operation by checking the dip switches located under the spaceball unit. To operate the spaceball with the code generated from this research the dip switch 2 must be in the closed position and the other switches must be in the open position. To operate the spaceball, the user must first compile and link the module into the main program. After the user spawns the *jtrackball* module he can move joint one of the manipulator. There are 8 function buttons on the DIM 6 spaceball, which when pressed will provide control to that joint. For instance if button number 5 is pressed then joint 5 is the controlled joint. Function buttons 7 and 8 are reserved to deactivate the spaceball module.

The DIM 6 can also control the PUMA manipulator in Cartesian space. The only difference from operating in this mode from the joint space mode is the requirement to spawn the *invkin* module. The DIM 6 spaceball allows a user to input all required inputs; therefore, only function buttons 7 and 8 are used to deactivate the module. Once the *ctrackball* is compiled and linked into the main program, the user spawns the *invkin* and *ctrackball* modules to control the manipulator.

Also implemented in the CHIMERA operating system was the Logitech mouse. The mouse is connected to the system by connecting the serial cable from the mouse to the processor card. No external connections are required. To operate the mouse, the user must compile and link the *cmouse* module into the main program then spawn the module. The mouse can only control two axis at a time. If the user presses the left mouse button, then Y and Z are the controlled axis. The middle button controls the X and Z while the right mouse button controls the X and Y axis.

I also used the Logitech mouse to produce a demonstration of shared control. The demonstration allows the user to control two dimensions while the manipulator is controlled to stay against a white board. This demonstration requires 3 modules. They are *invkin, jr3fts,* and *xforce.* The *jr3fts* module reads the JR3 force sensor and loads the state table with the appropriate force data. The *xforce* module computes the force feedback law and updates the x axis state table value. It also reads the input from the mouse and updates the Y and Z state table values. The user must compile all three modules and link them into the main program. To operate the modules the user must ensure the force sensor is started prior to any contact with any surfaces. The force sensor initializes itself upon start up and if there are unwanted forces applied then the sensor may give erroneous readings. After the user starts the *jr3fts* module, he should move the manipulator to the white board. Once the manipulator is near or touching the white board, he can start the *invkin* and *xforce* modules. The module only allows the user to control the Y and Z axis when the left mouse button is pressed. As the user moves the mouse the manipulator will follow in the Y and Z axis, and stay in contact with the white board.

IV.5 Shared Control

There are many interpretations of shared controlled. For this research the definition is a user and computer are combined to accomplish a task or movement. A good example of this definition is the demonstration I produced on the CHIMERA operating system. The demonstration allows the user to control the Y and Z axis while the computer controls the X axis. The code generated is a good start to allowing a user to control a paint gun or a paint stripping spray gun. In the proposed scenario, the user would control the spray gun by using a 2 DOF input device and the computer would control the distance spray gun is from the aircraft skin. This scenario gives the Air Force two advantages. First, the system

can use a standard inexpensive 2 DOF input device. Secondly, the intensive task of keeping the tool at a constant required distance is easily controlled by the computer while the human can use his or her cognitive skills to control the movement of the tool.

IV.6 Conclusion

Table 4.2 shows the grade of the input devices for each action. The highlighted cells show the input device with the best grade for that action. The first and fifth actions have the same input device as the optimal device. The optimal input device for these actions is the Ballpoint from MicroSoft. The reason for this is because the device is cheap, somewhat reliable, and has a button. Because the numbers were so close, the second most optimal input device was also highlighted which was the Logitech SpaceControlMouse. Actions two and three also have the same input devices as the optimal. P Q Control's 215 and 220 are the best because they are the cheapest device with a relatively high dexterity rating. The last two actions show the most dexterous input devices for two DOF and three DOF situations. It should be pointed out that dexterity can be a subjective rating. A user may like an input device better then another device and operate that device much better then another. With the open architecture that the Air Force's system will have, a user can pick his or her best device. The system would then handle calibrating that device into the architecture. If another user wants a different input device, then again the system should handle calibrating the different input device into the architecture.

Since there is more then one device for an action, it is recommended that each device that is optimal for an action be purchased and interfaced into the Air Force's generic telerobotics architecture. The actions are dividable within the task so the combination of the input devices will not be confusing nor difficult to understand. The user can use the P

Q Control's model 215 to move the manipulator's platform into position, then use the Schilling six DOF controller to perform the manipulator control. Using more then one input device to handle the task gives the user flexibility in controlling the entire task. The user can control the platform with one hand while controlling the manipulator with the other. There are also other combinations that can be used to enhance the overall system performance.

V. Conclusions and Future Work

This chapter will discuss the conclusions drawn from the thesis work accomplished. It will list the contributions of the research and discuss follow-on projects that stem from this thesis work.

V.1 Contributions

The contributions from this research to the robotics field are listed below. These contributions are specific in nature and are application driven. They can, however, be used in other such endeavors that require a selection process.

1. A methodology to select the optimal input device or manual controller for a generic telerobotics architecture was developed. This methodology will aid in selecting any device that is task driven as long a set of requirements can be derived for that task.
2. A set of requirements were derived given the specifications for the Air Force's generic telerobotics architecture. These requirements are application specific to the mobile platform architecture that seems to be the architecture of choice.
3. A scheme of shared control was briefly studied and revealed during this research. The shared control of the manipulator was devised that allowed the user to control two dimensions while the computer controlled the third. This scheme allows a user to strip paint or paint a large aircraft using an inexpensive two DOF input device.
4. Implementation issues were resolved in this research. Four input devices were interfaced to the Air Force Institute of Technology (AFIT) Robotics and Automation Applications Group (RAAG) system. The source code developed to interface the input devices will be delivered to Armstrong Lab (AL) and the Air Force's Robotics Center of Excellence (RACE). This source code will allow the Air Force to further study possible configurations for the generic telerobotics architecture.

V.2 Research Conclusions

It was discovered that if the task was broken into parts or actions, then there may be more then one "optimal" input device. Thus, should only one of the input devices be purchased and used for a complete system? The answer to that question is no; let the user decide. The system should be flexible enough to allow the user his or her own choice. If the user wants to use a three DOF joystick to move the scissors truck into place (Action 2, Chapter 4), and then wants to use a Schilling master controller to control the manipulator, then those options should be available. With the increase in computer power, object based programming, and flexibility in hardware, this user selected input device system is possible.

If system requirements prevents or does not lend itself to a more then one input device concept, then the actions for a particular task must be prioritized and the appropriate input device selected. It is well know that a human can adapt rather well to certain situations. This adaptation must be utilized if only one input device is permitted.

When all variables concerned are looked at to include size, weight, cost, etc., an "optimal" input device can be selected for a particular action. So many times in the past such surveys only included performance as a criteria. This applied approach does not place more emphasis on size, weight, cost etc., over that of performance. It only includes all possible variables that may be of concern to the over all system performance. This research also compared currently available devices. It did not develop a theoretical device where all variables were optimal.

This research also showed that shared control of a manipulator allows the user to accomplish what he or she does best, make decisions and control, and allows the computer to perform what it does best, maintain a set of variables and maintain accuracy. The shared control of the manipulator demonstrated that a manipulator can be controlled by an input device that has fewer dimensions than the task space and yet perform the task without any problems. It was shown in this research that a task space having three DOFs could be handled with a two DOF input device. This shared control has its advantages because there are a considerable amount of two DOF input devices available and the input devices are inexpensive. This control method also combines the best features of both the computer and the user. The user can control the manipulator through an unstructured environment of an aircraft skin while the computer maintains a constant force on the aircraft skin.

V.3 Follow-on Projects

There are several follow-on projects that can stem from this research. The first project is the completion of the Air Force's Generic Telerobotics architecture for large aircraft maintenance and repair. The second is to enhance CHIMERA to include a debugger, use ONIKA, a graphics programming tool, to control modules, and creating a robust inverse kinematics for the PUMA manipulator. The third project is to design and implement an open architecture necessary for robotics implementation in the Air Force. Another area that must be studied is the amount of shared control a manipulator should have at any given moment of task accomplishment. Another possible follow on project is a survey of the available real time operating systems and how they compare with each other.

The Air Force's generic telerobotics architecture will use the code generated throughout this research. The code generated during this effort will be delivered to the Robotics and Automation Center of Excellence (RACE) which is responsible for the generic telerobotics architecture development. RACE will use the code to aid in the human to machine interface and then will continue into other areas of concern.

CHIMERA has several areas that could be improved. One such area is a debugger. Currently, there is no debugger available, which can hindering the programmer when the code is not functioning properly. Another enhancement that should be pursued is with communicating with the outside world. CHIMERA has a library of Telnet commands that allows CHIMERA to establish a communications socket to another machine. The drawback is the other machine must be a Sun or compatible machine. If the CHIMERA library allowed any machine to connect to the CHIMERA socket, then an open architecture could be supported more easily.

Another project stemming from this research is the design and implementation of an open architecture. Should shared memory locations be used or message passing as the means to communicate between the open architecture modules? The answer probably does not have a definite yes or no solution. It will most likely be a combination of some sort.

Another project that should be developed is an autonomous verses shared control question. How much of the system should be controlled by the computer and how much by the human? The answer will be dynamic and will most likely be a function of task performance.

CHIMERA is not the only real time operating system available. We should survey the available operating systems that are available and compare them to each other. This kind of analysis would not only be valuable to the Air Force but also to any company wanting to operate in real time.

V.4 Conclusion

The conclusion drawn from this research is that no one device is the "optimal" device for a desired task. To accomplish a task such as stripping paint from a large aircraft, the user may desire more then one input device or manual controller. With the increase in computer power, object based programming, and flexibility in hardware, an on-the-fly user selected input device system is possible. Also, with the computer maintaining some of the users burden, a less capable input device can be used to accomplish a task. The shared control method allows the user to have control over the task while the computer maintains the accuracy. This combines the favorable characteristics of both the humans ability of control and cognitive decisions with the computers ability to maintain accuracy for a long period of time.

Bibliography

1. NASA Jet Propulsion Lab (JPL) Pasadena CA, A Generic Telerobotics Architecture for C-5 Industrial Processes. Prepared for Air Force Material Command (AFMC) Robotics and Automation Center of Escellence(RACE), San Antanio Air Logistics Center Kelly AFB, TX 78241.

2. H. Asada and J. K. Salisbury, Telerobotics and Applications. 1991

3. A. Bejczy, "Teleoperatoin: The Language of the Hand", IEEE International Workshop on Robot and Human Communication. 32-43. IEEE Press, 1992

4. A. Bejczy and J. Salisbury Jr., "Controlling Remote Manipulators Through Kinesthetic Coupling", Computers in Mechanical Engineering, 48-60, July 1983.

5. William R Corlis and Edwin G. Johnson, Teleoperators and Human Augmentation. Washington D.C., 1967

6. K. A. Farry and I. D. Walker, Myoelectric Teleoperation of a Complex Robotic Hand. 1993 IEEE International Conferenc on Robotics and Automation, IEEE Press.

7. Matthew W Gertz and Pradeep K. Khosla. ONIKA Iconic Programming Language and Human-Machine Interface. Carnegie Mellon University, 1994.

8. Walter W. Hankins III and Randolph W. Mixon, Baseline Experiments in Teleoperator Control. NASA Scientific and Technical Information Branch, 1986

9. R. D. Howe, A Force-Reflecting teleoperated hand System for the Study of Tactile Sensing in the Precision Manipulation, 1992 IEEE International Conference on Robotics and Automation, IEEE Press.

10. R. K. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen, "Integrality and Separability of Input Devices", ACM Transactions on Computer-Human Interaction, 3-26, March 1994.

11. H. Kazerooni and S. L. Mahoney, Dynamics and Control of Robotic Systems Worn by Humans. 1991 IEEE International Conferenc on Robotics and Automation, IEEE Press.

12. H. Kazerooni, Tsing-Ian Tsay and Karin Hollerbach, "A Controller Design Framework for Telerobotic Sysems", IEEE Transactions on Control Systems Technology, 50-62, 1993.

13. H. Kazerooni, "Human-Robot Interaction via the Transfer of Power and Information Signals", IEEE Transactions on Systems, Man, and Cybernetics, 450-463, 1990.

14. R. A. Lindemann. Construction and Demonstration of a 9-String, 6-DOF Force Reflecting Joystick for Telemanipulation. MS Thesis, University of Texas at Austin, 1987.

15. J. S. Lipscomb and M. E. Pique, "Analog Input Device Physical Characteristics", SIGHI, 40-45, July 1993.

16. I. S. Mackensie and W. Buxton, "A Tool for the rapid Evaluation of Input Devices Using Fitt's Law Models", SIGHI, 58-63, July 1993.

17. T. B. Malone, Teleoperator System, Man Machine Interface Requirements for Satellite Retrieval and Servicing. Essex Corporation, 1972

18. D. McAffee, "Diverse Applications of Advanced Man-Telerobot Interfaces", NASA, Technology 2000. 350-360, March 1991

19. "Human Machine Interfaces for Teleoperators and Virtual Environments", NASA CP 10071. NASA, 1990

20. Teloperator Human Factors Study Final Report. NASA CR 178930, 1986.

21. ESSEX Corporation, Human Operator Performance of Remotely Controlled Tasks: A Summary of Teleoperator Research Conducted at NASA's George C. Marshall Space Flight Center Between 1971 and 1981. Prepared for: NASA George C. Marshall Space Flight Center, N83-18251, March 1982.

22. NASA, Sensing and Force-Reflecting Exoskeleton. NASA Tech Brief, 98-99, October 1993.

23. M. Rahimi and W. Karwowski, Human-Robot Interaction. Taylor & Francis, 1992

24. Thomas B. Sheridan, Telerobotics, Automation,and Human Supervisory Control. The MIT Press Cambridge, MA, 1992

25. Ching-Shyong Shieh. A Puma 600 Robot Used as a Joystick Controller for Teleoperations. PhD Dissertation, University of Florida, 1990 Order Number 9121661.

26. David B. Stewart and Pradeep K. Khosla. CHIMERA 3.1 The Real-Time Operating System for Reconfigurable Sensor Based Control System. Carnegie Mellon University, 1993.

27. K. Willshire, W. Harrison, E. Hogge, R. Williams, and D. Soloway, "Results of Telerobotic Hand Controller Study Using Force Information and Rate Control", AIAA Space Programs and Technologies Conference. Amercan Institute of Aeronautics and Astronautics, AIAA 92-1451, March 1992

APPENDIX A

Table A.1 tabulates the input devices that will be analyzed. The first two columns contain the manufacture and model of the input device. The third column contains the power rating of the input device. A low number represents a good power rating (1 means no external power supply is required) and a high number represents a bad power rating. The fourth column describes the type of communication interface that is available with the input device. The possible choices are serial, parallel, analog, and other. The fifth column contains the number of DOFs the device can supply to a system. The next set of six columns describe the resolution of each available DOF. A number of one represents a resolution of infinity which means the signal is an analog signal and a number of ten represents a DOF resolution of one bit. A one bit resolution is another way of interpreting the output of a toggle switch. The last two columns contain the number of buttons on the input device and the type of input device. The possible choices for this column are joystick, mouse, spaceball, trackball, other. The column may contain a description of the device rather then the type.

Table A.1 Engineer Analysis

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
Applied Resources Corp.	Digital	5	Serial	3	6	6	6				0	Joystick
	Analog	5	Analog	3	1	1	1				0	Joystick
	Dual Gate	5	Switch	3	10	10	10				0	Joystick
Appoint	MousePen	1	Serial	2	4	4					2	Mouse
	Gulliver	1	Serial	2	4	4					2	Mouse
	Thumbelina	1	Serial	2	4	4					3	Mouse
Assmann Data Products	Digitus Magic Click	1	Serial	2	4	4					3	Mouse
CH Products	Switchstick	5	Switch	2	10	10					0	Joystick
	200 MK-III	5	Analog	2	1	1					1	Joystick
	400 MK-III	5	Analog	3	1	1					1	Joystick
	0 Inductive	5	Analog	2	1	1					0	Joystick
	1 Inductive	5	Analog	2	1	1					0	Joystick
	2 Inductive	5	Analog	2	1	1					0	Joystick
	4 Inductive	5	Analog	2	1	1					0	Joystick
	DT225	5	Serial	2	4	4					4	Trackball
	P150	5	Serial	2	5	5					0	Trackball
	P200	5	Serial	2	7	7					0	Trackball
CIS Graphic & Bildverarbeitung	DIM6	5	Serial	6	6	6	6	6	6	6	8	Spaceball

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
CTI Electronics Corp.	F1000-N2	5	Anolog	2	1	1					0	Joystick
	F1000-N5	5	Anolog	2	1	1					0	Joystick
	F2000-N2	5	Anolog	2	1	1					0	Joystick
	F2000-N5	5	Anolog	2	1	1					0	Joystick
	H0000-N5	5	Anolog	2	1	1					0	Joystick
	H0800	5	Anolog	3	1	1	1				0	Joystick
	H3003-N81	5	Serial	2	6	6					0	Joystick
	H8000-N81	5	Serial	2	6	6					0	Joystick
	H9000-N81	5	Serial	2	6	6					0	Joystick
	M1000-N24	5	Serial	2	6	6					0	Joystick
	M3000-N24	5	Serial	2	6	6					3	Joystick
	M4010-N24	5	Serial	2	6	6					3	Joystick
	M8003-N24	5	Serial	2	6	6					3	Joystick
Cyber-tech, Inc.	8900/500	5	Anolog	2	1	1					7	Joystick
	8600/500	5	Anolog	2	1	1					13	Joystick
	1800/500	5	Anolog	2	1	1					6	Joystick
	1200/500	5	Anolog	2	1	1					8	Joystick
	BRUT-88/500	5	Anolog	2	1	1					5	Joystick
	300-H/500	5	Anolog	2	1	1					3	Joystick
	8900/1050	5	Switch	2	10	10					7	Joystick
	8600/1050	5	Switch	2	10	10					13	Joystick
	1800/1050	5	Switch	2	10	10					6	Joystick
	1200/1050	5	Switch	2	10	10					8	Joystick
	BRUT-88/1050	5	Switch	2	10	10					5	Joystick
	300-H/1050	5	Switch	2	10	10					3	Joystick

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
GE Co.	PJJ31	5	Switch	1	10						0	Joystick
	PJB31	5	Switch	1	10						0	Joystick
	PJC53	5	Switch	2	10	10					0	Joystick
	PJD53	5	Switch	2	10	10					0	Joystick
	PJL96	5	Switch	2	9	9					0	Joystick
	PJM96	5	Switch	2	9	9					0	Joystick
	PJL97	5	Switch	2	9	9					0	Joystick
	PJM97	5	Switch	2	9	9					0	Joystick
Happ Controls, Inc.	50-6070	5	Switch	2	10	10					0	Joystick
	50-7608	5	Switch	2	10	10					0	Joystick
	50-6083	5	Switch	2	9	9					0	Joystick
	50-6090	5	Switch	2	9	9					0	Joystick
	56-5500	5	Serial	2	6	6					0	Trackball
	56-0100	5	Serial	2	6	6					0	Trackball
Hed, Hydro Electronics Devices Corp.	DAL-002	5	Anolog	2	1	1					0	Joystick
	DAL-006	5	Anolog	2*	1	1					0	Joystick
	DAL-007	5	Switch	2	10	10					0	Joystick
	DAL-008	5	Switch	2	10	10					0	Joystick
IZU Products Co.	MACFLY	1	Serial	2	6	6					2	Joystick
Kensington	Expert	1	Serial	2	8	8					2	Trackball
KeyTronic Corp.	Professional	1	Serial	2	7	7					3	Mouse
	Honeywell	1	Serial	2	7	7					3	Mouse
Kraft Systems Inc	Mouse	1	Serial	2	1	1					3	Mouse
	Trackball	1	Serial	2	3	3					3	Trackball

* Third DOF is optional.

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
Logitech	Mouse	1	Serial	2	3	3					3	Mouse
	Space Control Mouse	1	Serial	3	3	3	3				3	Mouse
	MouseMan	3	Serial	2	3	3					3	Mouse
	Trackman	1	Serial	2	3	3					3	Trackball
	WingMan	1	Serial	2	3	3					8	Joystick
Machine Components Corp.	31MX	5	Switch	1	10						0	Joystick
	51MX	5	Switch	1	9						0	Joystick
	71MX	5	Switch	1	9						0	Joystick
	51MXY	5	Switch	2	10	10					0	Joystick
	51DXY	5	Switch	2	10	10					0	Joystick
	91MO	5	Switch	4**	10	10	10	10			0	Joystick
	91MXY	5	Switch	2	9	9					0	Joystick
	91DMXY	5	Switch	2	9	9					0	Joystick
	91MDXY	5	Switch	2	9	9					0	Joystick
	52MXY	5	Switch	2	10	10					0	Joystick
	52DXY	5	Switch	2	10	10					0	Joystick
	171MO	5	Switch	4	9	9	9	9			0	Joystick
	92MXY	5	Switch	2	10	10					0	Joystick
	92DMXY	5	Switch	2	10	10					0	Joystick
	92MDXY	5	Switch	2	10	10					0	Joystick
	54MXY	5	Switch	2	10	10					0	Joystick
	54DXY	5	Switch	2	10	10					0	Joystick
	41MXZ	5	Switch	2	10	10					0	Joystick
	61MXYZ	5	Switch	3	10	10					0	Joystick
	H51MXY	5	Switch	2	10	10					0	Joystick

** Labeled as 4 DOF but is really a 2 DOF joystick with contacts at 45 degrees

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
Maurey Instrument Corp.	JSP	5	Anolog	3	1	1	1				Optional	Joystick
	JSE	5	Anolog	2	1	1					0	Joystick
	SJS2491	5	Switch	2	10	10					0	Joystick
	SJS2490	5	Switch	2	10	10					0	Joystick
	HDJ	5	Anolog	2	1	1					0	Joystick
Measurement Systems, Inc.	462	5	Anolog	2	1	1					0	Joystick
	469	5	Anolog	2	1	1					0	Joystick
	467	5	Anolog	2	1	1					0	Joystick
	465	5	Anolog	2	1	1					0	Joystick
	470	5	Anolog	2	1	1					0	Joystick
	463	5	Anolog	3	1	1					0	Joystick
	473	5	Anolog	3	1	1	1				0	Joystick
	446	5	Anolog	2	1	1					4	Joystick
	435	5	Anolog	2	1	1					0	Joystick
	521	5	Anolog	2	1	1					0	Joystick
	531	5	Anolog	2	1	1					0	Joystick
	523	5	Anolog	1	1						0	Joystick
	546	5	Anolog	2	1	1	1				4	Joystick
	547	5	Anolog	3	1	1	1				4	Joystick
	402	5	Anolog	2	1	1					4	Joystick
	570	5	Anolog	2	10	10					0	Joystick
	575	5	Anolog	2	10	10					0	Joystick
	615	5	Anolog	2	6	6					0	Trackball
	625	5	Anolog	2	5	5					3	Trackball
	626	5	Anolog	2	5	5					0	Trackball
	STX	5	Serial	2	3	3					0	Joystick
	JTX	5	Serial	2	3	3					3	Joystick
	XLT	5	Serial	2	6	6					3	Trackball

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
Merit, J.R. Controls Inc.	NSO	8	Anolog	2	8	8					0	Joystick
	NS2	8	Anolog	2	8	8					0	Joystick
	CSO	8	Anolog	2	8	8					0	Joystick
	CS2	8	Anolog	2	5	5					0	Joystick
	MO	8	Anolog	2	1	1					0	Joystick
	M2	8	Anolog	2	10	10					0	Joystick
	SL-M0	8	Anolog	2	1	1					0	Joystick
	VDF	8	Anolog	2	1	1					0	Joystick
Microsoft Corp.	Mouse	1	Serial	2	3	3					2	Mouse
	BallPoint	1	Serial	2	4	4					2	Trackball
MicroSpeed Inc.	PC-Trac	1	Serial	2	1	1					2	Trackball
	Micro-Trac	1	Serial	2	2	2					3	Trackball
Mouse Systems Corp.	NewMouse	1	Serial	2	3	3					3	Mouse
	OmniMouse	1	Serial	2	3	3					2	Mouse
	PC Mouse III	1	Serial	2	3	3					3	Mouse
	PC Mouse	1	Serial	2	3	3					3	Mouse
	PC Mouse 3D	1	Serial	3	3	3	3				3	Mouse
	WhiteMouse	1	Serial	2	4	4					3	Mouse
	PC Trackball	1	Serial	2	3	3					2	Trackball
OEM Controls, Inc.	JS1	5	Analog	2	1	1					0	Joystick
	JS2	5	Analog	2	1	1					0	Joystick
	JS5	5	Analog	2	1	1					0	Joystick
	JS6	5	Analog	2	1	1					0	Joystick
	MS2	5	Analog	1	1						0	Joystick
	MS4	5	Analog	1	1						0	Joystick
Polar & Pole Inc.	ATB600	1	Serial	2	1	1					3	Trackball
	ATB1200	1	Serial	2	1	1					3	Trackball
Prohance Technologies	Mouse	1	Serial	2	4	4					2	Mouse
	Trackball	1	Serail	2	4	4					3	Tackball

Table A.1 Engineer Analysis cont'd

Company Name	Model	Power	Comm. Interface Type	DOFs	Resolution per DOF						Buttons	Type
PQ Controls Inc	215	5	Anolog	3	1	1	1				2	Joystick
	220	5	Anolog	3	1	1	1				2	Joystick
Sauer-Sundstrand Electronics Systems	MCH	8	Anolog	1	1						3	Joystick
Sumtak Encoders	STB	5	Other	2	4	4					0	Trackball
Suncom	ICONtroller	1	Serial	2	5	5					4	Joystick
Alternative Input Devices												
Immersion Human Interface Corp.	PROBE-IC	8	Serial	3	3	3					0	Other
	PROBE-IX	8	Serial	3	2	2					0	Other
	PROBE-MD	8	Serial	3	1	1					0	Other
Schilling Development Inc.	TITAN	5	Serial	6	2	2	2	2	2		13	Kinematic Matched
Spectra Symbol Corp.	MEMBRANE	5	Serial	2	2	2					20	Membrane

APPENDIX B

Appendix B describes the input devices as a user would look at them. It will tabulate such things as cost, reliability, dexterity, size, and weight. Table B.1 shows the results of the survey tabulation. The first two columns of this table contain the company that manufactures the input device and the model of the input device. The third column contains the cost of the input devices. The cost is divided into 20 parts. A number of one represents a cost that is under one hundred dollars. A ten represents a cost of at least $10,000. The divisions are not linear so that a wider range of dollar values could be rated. Table 3.2 on page 3-7 fully describes the cost divisions. The fourth column contains reliability. A one in this column represents a very reliable device and a ten represents a almost assured failure when the input device is used. The fifth column contains dexterity. The size and total displacement of the input device was used to develop a rating of dexterity. The number one represents a dexterous device while a ten represents considerable difficulty in operating the input device. The sixth and seventh columns describe the size and weight of the input device, respectfully. A number of one represents the smallest and lightest possible device out of all possible input devices. A number of ten represents the heaviest and biggest. The last column is self-centering. A Y in this column means the input device will return to its original position when released which implies the device is a non-displacement device. An N in this column means the input device will not return to its original position.

Table B.1 User Analysis

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
Applied Resources Corp.	Digital	3	5	4	Finger	6	4	Y
	Analog	3	6	4	Finger	3	3	Y
	Dual Gate	2	5	4	Finger	3	3	Y
Appoint	MousePen	1	7	8	Pen	1	1	N
	Gulliver	1.5	7	8	Finger	1	1	N
	Thumbelina	1	7	7	Mouse	1	1	Y
Assmann Electronic Inc.	Digitus Magic Click	1	6	6	Mouse	1	1	Y
C H Products	200 MK-III	1.5	5	4	Finger	3	3	Y*
	400 MK-III	1.5	5	4	Finger	3	3	Y
	0 Inductive	1.5	10	4	Finger	3	3	Y
	1 Inductive	1.5	10	4	Finger	3	3	Y
	2 Inductive	1.5	10	4	Finger	3	3	Y
	4 Inductive	1.5	10	4	Finger	3	3	Y
	DT225	1.5	5	6	Ball	4	3	Y
	P150	1.5	5	6	Ball	4	3	Y
	P200	1.5	5	6	Ball	4	3	Y
CIS Graphic & Bildverarbeitung	DIM6	3.5	10	9	Full Ball	7	6	Y

* CH Products' joysticks have a patented system where as the joystick can become a self-centering or a non self-centering joystick.

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
CTI Electronics Corp.	F1000-N2	1.5	3	6	Finger	4	NDA**	Y
	F1000-N5	2	3	6	Finger	4	"	Y
	F2000-N2	2	3	6	Finger	4	"	Y
	F2000-N5	2	3	6	Finger	4	"	Y
	H0000-N5	2	3	6	Finger	4	"	Y
	H0800	2.5	3	5	Finger	4	"	Y
	H3003-N81	2.5	3	5	Finger	4	"	Y
	H8000-N81	2.5	3	5	Finger	3	"	Y
	H9000-N81	2.5	3	5	Finger	3	"	Y
	M1000-N24	2	2	6	Finger	4	"	Y
	M3000-N24	2	2	6	Finger	4	"	Y
	M4010-N24	2	2	6	Finger	4	"	Y
	M8003-N24	2	2	6	Finger	4	"	Y
Cyber-tech, Inc.	8900/500	2	8	2	Hand Trigger	5	6	Y
	8600/500	2	8	2	Hand Trigger	5	6	Y
	1800/500	3	8	2	Hand Trigger	5	6	Y
	1200/500	3	8	2	Hand Trigger	5	6	Y
	BRUT-88/500	2	8	2	Hand Trigger	5	6	Y
	300-H/500	2	8	2	Hand Trigger	5	6	Y
	8900/1050	2	8	2	Hand Trigger	5	6	Y
	8600/1050	2	8	2	Hand Trigger	5	6	Y
	1800/1050	2	8	2	Hand Trigger	5	6	Y
	1200/1050	2	8	2	Hand Trigger	5	6	Y
	BRUT-88/1050	2	8	2	Hand Trigger	5	6	Y
	300-H/1050	2	8	2	Hand Trigger	5	6	Y

** No Data Available for this item.

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
GE Co.	PJJ31	2	5	8	Finger	2	3	N
	PJB31	2	5	8	Finger	2	3	Y
	PJC53	2	5	8	Finger	2	3	N
	PJD53	2	5	8	Finger	2	3	Y
	PJL96	4	5	8	Finger	2	3	N
	PJM96	4	5	8	Finger	2	3	Y
	PJL97	4	5	8	Finger	2	3	N
	PJM97	4	5	8	Finger	2	3	Y
Happ Controls, Inc.	50-6070	1	5	7	Finger	5	4	Y
	50-7608	1	5	7	Finger	5	4	Y
	50-6083	1	5	7	Finger	5	4	Y
	50-6090	1	5	7	Finger	5	4	Y
	56-5500	2	5	7	Finger	5	4	N
	56-0100	2	5	7	Finger	5	4	N
Hed, Hydro Electronics Devices Corp.	DAL-002	3	1	5	Finger	3	4	Y
	DAL-006	3	1	3	Hand/Hand Trigger	4	4	Y
	DAL-007	3	5	5	Finger	3	4	Y
	DAL-008	3	5	5	Finger	3	4	Y
IZU Products Co.	MACFLY	1	8	4	Hand Trigger	5	4	Y
Kensington	Expert	2	6	6	Ball	4	4	Y
KeyTronic	Proffesional	2	6	6	Mouse	4	2	Y
	Honeywell	1	7	6	Mouse	4	2	Y
Kraft Systems Inc	Mouse	1	7	6	Mouse	5	3	Y
	Trackball	1	7	6	Ball	4	4	Y
Logitech	Mouse	1	6	6	Mouse	5	2	Y
	Space Control Mouse		6	6	Mouse	3	2	Y
	MouseMan	1.5	9	6	Mouse	3	2	Y
	TrackMan	1	6	6	Ball	5	3	Y
	WingMan	1	6	6	Double Hand	6	3	Y

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
Machine Components Corp.	31MX	1.5	4	8	Finger	3	2	Y
	51MX	1.5	4	8	Finger	3	2	N
	71MX	1.5	4	8	Finger	3	2	Y
	51MXY	2	4	8	Finger	3	2	Y
	51DXY	2	4	8	Finger	3	2	N
	91MO	2	4	8	Finger	3	2	Y
	91MXY	2	4	8	Finger	3	2	Y
	91DMXY	2	4	8	Finger	3	2	Y/N***
	91MDXY	2	4	8	Finger	3	2	N/Y
	52MXY	2	4	8	Finger	3	2	Y
	52DXY	2	4	8	Finger	3	2	N
	171MO	2	4	8	Finger	3	2	Y
	92MXY	2	4	8	Finger	3	2	Y
	92DMXY	2	4	8	Finger	3	2	N/Y
	92MDXY	2	4	8	Finger	3	2	Y/N
	54MXY	2	4	8	Finger	3	2	Y
	54DXY	2	4	8	Finger	3	2	N
	41MXZ	2	4	8	Finger	3	2	Y
	61MXYZ	2	4	8	Finger	3	2	Y
	H51MXY	2	4	8	Finger	3	2	Y
Maurey Instrument Corp.	JSP	2.5	6	7	Finger	4	3	Y
	JSE	2.5	2	7	Finger	4	3	Y
	SJS2491	2	6	7	Finger	4	3	Y
	SJS2490	2	6	7	Finger	4	3	Y
	HDJ	1.5	2	7	Finger	4	3	Y

*** The set of devices have more then one contact per DOF. Some contacts are self-centering and others are not. Starting from the center of the device, the first contact is self-centering based on the answer in this column. The next contact in the same DOF is self-centering or not based on the second answer in this column.

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
Measurement Systems, Inc.	462	1.5	2	8	Force Finger	1	1	Y
	469	3	2	8	Force Finger	1	1	Y
	467	2	2	8	Force Finger	1	1	Y
	465	2	2	8	Force Finger	1	1	Y
	470	2	2	8	Force Finger	1	1	Y
	463	2	2	8	Force Finger	1	1	Y
	473	2	2	8	Force Finger	1	1	Y
	446	2	2	8	Force Finger	1	1	Y
	435	2	2	8	Force Finger	1	1	Y
	521	2	2	8	Finger	2	2	Y
	531	2	2	8	Finger	2	2	Y
	523	2	2	8	T Handle	5	5	Y
	546	2	2	4	Hand	5	4	Y
	547	2	2	4	Hand	5	4	Y
	402	2	2	5	Force/ Displacement Hand	5	4	Y
	570	2	6	8	Finger	2	2	Y
	575	2	5	4	Hand	5	3	Y
	615	2	5	6	Ball	5	4	Y
	625	2	5	6	Ball	5	4	Y
	626	2	5	6	Ball	5	4	Y
	STX	2	2	8	Finger	2	1	Y
	JTX	2	2	8	Finger	5	4	Y
	XLT	2	2	6	Ball	5	5	Y

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
Merit, J.R. Controls Inc.	NSO	1.5	1	2	Hand	10	10	Y
	NS2	2	1	2	Hand	10	10	Y
	CSO	2	1	2	Hand	10	10	Y
	CS2	2	1	2	Hand	10	10	Y
	MO	2	1	2	Hand	5	6	Y
	M2	2	1	2	Hand	5	5	Y
	SL-M0	2	1	2	Hand	5	5	Y
	VDF	1.5	1	2	Hand	5	5	Y
Microsoft Corp.	Mouse	1	6	6	Mouse	5	4	Y
	BallPoint	1.5	6	6	Ball	2	1	Y
MicroSpeed Inc.	PC-Trac	1.5	7	6	Ball	4	3	Y
	Micro-Trac	1	6	6	Ball	2	1	Y
Mouse Systems Corp	NewMouse	1	6	6	Mouse	3	3	Y
	OmniMouse	1	6	6	Mouse	3	3	Y
	PC Mouse	1	6	6	Mouse	3	3	Y
	PC Mouse III	1	6	6	Mouse	3	3	Y
	PC Mouse 3D	2.5	6	6	Mouse	3	3	Y
	WhiteMouse	1	6	6	Mouse	3	3	Y
	PC Trackball	1.5	6	6	Ball	2	3	Y
OEM Controls, Inc.	JS1	2	2	6	Hand/Knob	5	7	Y
	JS2	2	2	6	Hand/Knob	6	7	Y
	JS5	2	2	6	Hand/Knob	4	7	Y
	JS6	2	2	6	Hand/Knob	5	7	Y
	MS2	2	2	6	Hand/Knob	5	7	Y
	MS4	2	2	6	Hand/Knob	5	7	Y
Polar & Pole Inc.	ATB600	2	6	6	Ball	3	1	Y
	ATB1200	2	6	6	Ball	1	1	Y

Table B.1 User Analysis cont'd

Company Name	Model	Cost	Reliability	Dexterity	Grip	Size	Weight	Self Cent.
Prohance Technologies	Mouse	1.5	7	6	Mouse	3	2	Y
	trackball	1	7	6	Ball	3	2	Y
PQ Controls Inc	215	1	2	3	Hand	5	4	Y
	220	1	2	3	Hand	5	4	Y
Sauer-Sundstrand Electronics Systems	MCH	1.5	2	5	Hand	5	7	Y
Sumtak Encoders	STB	1.5	3	6	Ball	3	2	Y
Suncom	ICONtroller	1	5	7	Finger	3	2	Y
Alternate Input Devices								
Immersion Human Interface Corp.	PROBE-IC	5.5	3	1	Stylus	6	6	N
	PROBE-IX	6	3	1	Stylus	6	6	N
	PROBE-MD	6.5	3	1	Stylus	6	6	N
Schilling Development Inc.	TITAN	9.5	2	2	Kenimatic Match	6	8	N
Spectra Symbol Corp.	MEMBRANE	1.5	1	7	Membrane	4	5	Y

APPENDIX C

C.1 Company Summaries

This appendix is divided into two parts. The first part is the joysticks, mice, and trackballs part that basically discuss the companies that manufacture either or all of the listed devices. The second part is concerned with any company that manufactures input devices that are alternative to the joystick, mouse, or trackball.

C.1.1 Joysticks, Mice, and Trackballs

This section will review the companies that supply input devices. The history of these companies and some of their other products will be discussed.

Advanced Gravis is a computer interface manufacturer that makes the Phoenix Flight & Weapons control system. The Phoenix is a complete flight simulator interface. It has the standard joystick and a weapon controller. Advanced Gravis also manufactures the MouseStick II and the Mac GamePad.

Applied Resources Corporation has been in business since July 1972 with electro-mechanical switches and manual/semi-automatic test equipment as the main products produced. Hermetically sealed safety/separation switches for the Tomahawk Cruise Missile is just one product of many that is produced by Applied Resources Corporation. Other products produced are environmental and hermetically sealed rotary, thumbwheel,

toggle, lanyard, remote, and hard link switches. Applied Resources Corporation also designs and manufactures specialized test equipment and machinery. Applied Resources can make any reliable custom input devices (joysticks). Applied resources Corporation sent some drawings of three joysticks they designed in the past. The drawings did not clearly show some the required data for complete analysis.

Appoint manufactures several input devices for the computer. Some of their most recent releases are the MousePen, Thumbelina, and Gulliver. Gulliver is an upgraded mouse pen that is small and for the small hand is easy to use. Thumbelina is probably the smallest trackball I've seen. It measures 1.6 inches in square and has three buttons.

Assmann Electronics Inc. manufactures the Digitus Magic Click. Assmann Electronics believes that the buttons on a mouse should be located such that the thumb and not the index finger operates them. If the user wants to still use the index finger, Assmann Electronics gave the user the option of either still by simply moving a switch located at the rear of the device.

CH Products manufactures industrial and commercial positioning devices. The joysticks have many options that cannot be all analyzed in the data table. Thus a few joysticks with different options will be chosen. CH Products have five basic joysticks which three of them can be tailored to fit a specific need. The three configurable joysticks are labeled the standard, miniature, and compact. The two non-configurable joysticks are called the switchstick and inductive. The configurable joysticks have several options. The first of many options are the model or the number of DOFs. A model 100 corresponds to a two DOF joystick, a model 200 is the same as a model 100 with an additional button on the end of the grip. A model 300 is a three DOF joystick and a model 400 is a model 300

with a button on the end of the grip. After choosing a model, the mounting bezel, housing, grip, and potentiometer must all be chosen. The data analysis tables contain four joysticks from CH Products. The switchstick and inductive joysticks are included as well as a model 200 (two DOF with button) and a model 400 (three DOF with button). No housings were included with the these joysticks so the size of the joysticks were as small as possible.

CH Products produces three trackballs with different options of protocol, mounting position, resolution, and baud rate. For the purpose of analysis all options were chosen to maximize the trackball performance. The first two models are almost identical with the only difference of the balls are a different size. The P150 contains a 1.5 inch ball where the P200 contains a two inch ball. The DT225 contains a 2.25 inch ball and hasfourbuttons that the other two do trackballs do not have. The nice thing about these trackballs are they are ready to go with their many different interface protocols. No interface circuit is required, just plug them in and let the software do the rest.

CH Products also produces the flight stick and the virtual pilot. These two devices are designed for the home PC game players. The Flightstick is a joystick that has a trigger hand grip with an additional button on the top. The virtual pilot is a pilot yoke simulator. It looks and (according to brochure) feels like a real airplane yoke. It has two buttons, one on each hand grip, that are easy to get at with the thumbs. It also has a trim and throttle control. These type of devices should not be quickly discarded as devices that would not work in an industrial system. They may provide a unique interface that a user may like.

CNS Inc. manufactures the Sicos Colani mouse. The mouse, input device for a PC or Macintosh, was designed by a European designer of airplanes and cars. It is ergonomically correct to fit the human hand. It has two buttons.

CTI Electronics Corporation produces joysticks, trackballs, and touch screens for harsh environments. CTI has several base models with eight different grips. Their biggest selling point is the reliability of their products. Numbers such as one million operating hours were mentioned in their brochure. CTI also produces trackballs which were not include in the analysis because no information was received describing the trackballs.

CyberTech inc. produces two base models that hold their six different hand grips. From the brochure I got the feeling that the grips are their main product with the joystick bases as only something to aid their grips.

GE Co. produces momentary and maintained contact joysticks or as referred to earlier as switch toggle joysticks. GE has three different types of joysticks which are classified by the number of switch locations. Each type of joystick can be ordered in a momentary activation (self centering) or a maintained activation (non-self centering) modes.

HAPP Controls inc. produces four joystick and one trackball as well as several different kinds of push buttons, an optical gun, and coin meter equipment. The joysticks are operated by either mechanical switches or photo electronic switches. HAPP Controls also produces a trackball/trackball interface. The interface allows up to three push buttons to be added to the trackball system. Since the push buttons must be ordered separately, they were not included in the analysis.

Hydro Electronic Devices, Inc (HED) produces joysticks, valve drivers, on/off and proportional remote controls, hydraulic controllers, and microprocessor based single board controllers. HED produces two very reliable inductive and two switch toggle joysticks.

IBM Corp. manufactures the IBM PS/2 Trackpoint. The Trackpoint is a mouse and trackball in one device. As a trackball it has four buttons and a one inch ball, flip it over, and the device becomes a mouse. The ball protrudes from the device more then a standard mouse thus there is some extra wobble.

Itac Systems Inc. manufactures the Mouse-trak which is a trackball that emulates a mouse. A user's wrist rests on a cushioned pad while the fingers activate the two or three programmable buttons and rotate a two-inch polished trackball. The trackball movement controls the cursor, and its movement sensitivity is adjustable. The input buttons functions are user-definable to support various software packages and different user preferences. Models range from Quadrature output, Serial, BUS and the PS/2 mouse port interface.

IZU Products makes the MACFLY joystick. The MACFLY joystick is specifically made to interface to a Macintosh computer. The joystick also comes with software to aid in installing and setting up the joystick. If desired the joystick also includes a game called hellcats over the pacific for $15.00.

Kensington Microware manufactures the expert mouse trackball. The expert mouse trackball is a trackball that is relatively small and has two buttons. The buttons can be changed from left hand to right hand by changing a switch in the back of the device.

KeyTronic Corp. manufactures the Honeywell designed Honeywell Mouse. The Honeywell mouse has two discs instead of a typical ball. The two discs are beveled and operated in opposing axis'. KeyTronic also manufactures the professional series mouse. This mouse unlike the Honeywell mouse is more pleasing to the eyes. It has a more contoured shape and has raised dots on the left button to aid the user in finding the correct button quickly.

KRAFT Systems manufactures a mouse and trackball for computers. Kraft also manufactures the Thunderstruck for flight simulators.

Logitech is unarguably the leader in computer input devices with the Logitech mouse outselling all other competitors. Logitech manufactures two DOF, three DOF, and cordless mice as well as trackballs and joysticks.

Machine Components Corp. (MCC) produces many industrial items such as clutches, brakes, couplings, indexing devices, spur gears, and toggle switch joysticks. The joysticks range from a single DOF to three DOF, self centering and non self centering, single pole or double pole, and even some locking joysticks. The locking joysticks remain in the locked off position until the user pulls the finger grip in an upward fashion. The joysticks can be ordered with either gold or silver contacts, phenolic or diallyl-phthalate casing, and many different current ratings. They can also come with many different bat handles orfourdifferent styles of boots. The model H51MXY uses hall effect switch elements that give it an excellent signal stability.

Maurey Instrument Corp. produces potentiometers, motorized potentiometers, single axis joysticks, and two DOF joysticks. The single axis joystick is a potentiometer with a handle on the stator. The signals for the two DOF joysticks are generated by potentiometers, inductors, or switches. Maurey Instrument Corp. produces both wirewound and conductive plastic types of potentiometers. The potentiometers can be ordered with resistance from 50Ω to 100KΩ and with sizes from .5 inches to almost three inches. The joysticks also come in a variety of grips, boots, and push buttons.

Measurement Systems Inc. (MSI) produces the only found joysticks that operates by reading the input forces. This company seems to have dedicated itself to the human to machine interface arena. They produce all kinds of input devices such as force operated joysticks, displacement joysticks, force/displacement joysticks, very small joysticks, switch toggle joysticks, trackballs, and all the necessary equipment to interface those devices to almost any computer or application. Some of their joysticks are very small (sizes of .5 by one inch are shown in the catalogue) and the weight is also very low (weights of about one ounce). These joysticks have a definite niche in the input devices world of the smallest joysticks available. MSI has mounted their miniature joysticks in several different grips which can then be mounted on one of the displacement joysticks that they also produce. MSI also makes four trackballs each with a different size ball. The three models labeled JTX, XLT, and STX are complete systems that are ready to interface to any computer through either a serial port or a quadrature square wave output signal.

J. R. Merrit Controls, Inc. produces heavy duty joysticks are constructed with one thing in mind: high power and reliability. This is the only company that means heavy duty when stated in the catalog. The NS2 is rated at 25 Amps, 600 volts AC and five amps at 250 volts. J. R. Merrit has definitely filled the niche of high power systems with their

joysticks. All the joysticks come in varies sizes and styles. For example the NSO can be ordered in one, two, or three DOF configuration if desired. It can also have different resolution depending on the application. J. R. Merrit also produces rotary switches, deadman switches, grips for joysticks, foot pedals, and control chassis'. The control chassis can have several joysticks, buttons, or other array of objects mounted on it. J. R. Merrit also produces a heavy duty arm chair that contains two control chassis on either arm. Bottom line on this company: if you want high power control devices then J. R. Merrit is the right company.

Microsoft Corp. manufactures several mice for a computer. The ball-point mouse is a small trackball that mounts on the side of a laptop. Microsoft is also the leading producer of the PC mouse.

MicroSpeed Inc. manufactures the PC-Trac trackball for the PC. This is the second edition of the PC-Trac which has some minor modification form the earlier Fast Trap. The trackball has two large buttons that encircle the ball. This give the user the option of pressing the buttons from under the ball or over the ball. MicroSpeed also manufactures the Micro-Trac. The Micro-Trac is a trackball for the laptop.

Mouse Systems Corp. manufactures several mice and trackball for the PC. The NewMouse is an optical mechanical mouse that looks like the PC mouse 3D without the 3D extras. The PC mouse 3D is a mouse that uses two side buttons in combination to make a three DOF input device. The OmniMouse like the PC mouse and the PC mouse II are a basic mouse with no frills. The whiteMouse is again a no frills mouse that gives the user a little more resolution over the other mice.

OEM Controls Inc. manufactures electrohydraulic valve controllers which includes several different models of joysticks, rotary switches, and foot pedals. The joysticks can be ordered with a solid ball, solid cap, two piece cap, or rocker grip handles. It can also have a self centering mechanism, friction hold, or a maintained mechanism. The joysticks can operate with either switches or potentiometers. OEM states in their brochure they can custom configure their devices to user specifications. They also manufacture a grip that has several buttons such as triggers, up to four thumb buttons, and optional rotation devices. Available to all models is the UFO electronics package, which is an interface from the potentiometers of the joysticks to a electrohydraulic proportional valve.

Polar & Pole inc. (P&P) manufactures the Agiler trackball and the handy trackball. The trackball has several different interface modes while the handy trackball operates only in a Microsoft mouse mode. Two big advantages to these input devices are there high resolution, up to 2900 dpi, and neither require any external power.

P Q Controls Inc. manufactures two models of joysticks. The model 215 and model 220 can be ordered with several options. There are three grip options, four DOF options, and three sensing systems. The highest quality option is the inductively coupled sensor.

Prohance Technologies manufactures a mouse and a trackball. the mouse id a mechanical mouse with two buttons. The trackball is a right handed device that has three buttons mounted to the right side of the ball. this layout indicates the ball should be used with the thumb and the three first fingers will operate the buttons.

Sauer Sundstrand manufactures a single axis control handle or a one DOF joystick. the joystick can have many options. From the base of the joystick to type of grip to sensors

are all optional. the joystick is made to interface to Sauer Sundstrand valves specifically but are flexible enough to use in other systems.

SUMTAK manufactures many items such as encoders, generators, limit switches, and a trackball. The trackball is a standard no thrills reliable device. It sends a pulse signal for each of the four direction.

Suncom Technologies manufactures the ICONtroller and the joystick FX2000. The ICONtroller is a small joystick that Velcro's to the side of a laptop or other computer. The FX200 is a game joystick with a hand grip and suction cups to mount on the top of any desk. Other joysticks made by Suncom are the Analog Plus, G-Force Flight Yoke, and Night Force.

Thrustmaster is known for its flight simulator input devices. One such input device is the Flight Control System (FCS) Mark1 which was integrated into the AFIT CHIMERA system. Thrustmaster has several other devices such as the F-16 FCS, Formula T1, Pro FCS, and others..

Z-NIX manufactures mice for the PC. One of there better known devices is the cordless super mouse. There is little information concerning this mouse at this time other then it is cordless, so the mouse was not compared to the other input devices.

C.1.2 Alternative Input Devices

BioControl Systems Inc. manufactures the Biomouse which is an eight channel biosignal processing platform that signal processes and maps a user's muscle, eye movement, or

brain signals to digital control code. A user can program or configure the system to control digital interface devices directly from the nerve signals. the price for this system is $15,000.

Carroll Touch Inc. manufactures the Touch System which is an overlay system that provides IBM PS/2 Model 8513 color monitors with touch input capabilities. It is designed for menu-driven applications such as business graphics or for customized turn-key packages. The system uses analog resistive technology, which utilizes a metallic-coated glass base layer and a flexible metallic-coated top overlay sensor. Each unit includes a cable and controller.

Cirque manufactures the GlidePoint which is a miniature touch-screen that attaches anywhere on your laptop or desktop system using Velcro. To point at a location on the screen, you simply glide your finger along the touch-screen to the location where you want the cursor to move on your system. To "click" on that location you tap your finger at that point on the touch-screen. To drag an object, you double-tap at that point and then glide your finger.

Communication Intelligence Corp manufacture the MacHandwriter or HandWriter for windows. A tablet and pen is used to supply the computer with a X and Y position. Clicking is accomplished by pressing the pen down.

Computability manufactures the AID+ME which is an access interface that helps the user to select a variety of entry methods/devices such as a membrane keyboard, switch, touch window, mouse or joystick. It is supported for the PC and PS/2 family of computers. Features include scanning setup, key redefinition, mouse emulation, voice output with on-

board synthesizer. It enables the user to run a variety of applications including text programs, educational software and CAD applications.

Creative Technology manufactures several 3D interactive devices that interface to a computer through their receiver unit. The three devices currently on the market are the AeroPen, AeroMouse, and AreoDuet. These three devices operate using the patented FreePoint infrared technology which gives the computer an X, Y, and Z position of the device. The user can manually select the virtual space and performance characteristics of the devices.

Digital Image Design Inc. manufactures the Cricket which is a 3D interactive device featuring upright orientation with several buttons. The thumb button on the Cricket allows the user to input two DOF direction. A variable vibration provides the user with tactile feedback.

Global Devices manufactures the Global 3D Controller which translates fullthreeDOF input from the users hand. The Controller, which costs $249, also includes 32 levels of active tactile feedback.

Greenleaf Medical Systems manufactures the Dataglove which collects data dynamically in 3D space through digital sensors located on the users hand on a lycra glove. The Dataglove was originally designed for medical applications.

GRiD Systems Corp. Manufactures the IsoPoint device. The IsoPoint is a device that is mounted directly into a laptop computer. It is straw shaped object that allows the user to input x and y position commands by rotating the straw shaped object or by sliding the

shaft back and forth. The actual device is very small but the only way to get the device, currently, is to order a complete laptop case. The IsoPoint does not come as a separate device which is a definite drawback. The cost of the IsoPoint is currently $5,095.00, which includes the laptop case and a SX computer.

Handykey Corp. manufactures the Twiddler which is an alternative keyboard. It is help by the user in the palm of the hand and with the combinations of different buttons the user can input any character which is on the keyboard. The cost of this device is $199.

Interlink manufactures the Propoint which is a handheld mouse that allows the user freely walk around during a presentation. This device costs $129 and is include with the IBM's 360 and 755 series thinkpad traveling Multimedia computers.

Immersion Human Interface Corp. produces a unique input device that does not fit any categories of usual input devices. The Immersion probe is a three DOF device with a pen like stylus as the grip. It also comes in three different models. The first being the cheapest and not so accurate while the third is the most expensive and has the best accuracy. The video sent by Immersion showed the immersions probe being used in several different applications. Immersion also produces several devices that could be used with the probe. For instance, foot pedals, thumb switches, and rotary knobs.

ISCAN manufactures the OPTIMOUSE which is a remote cursor-control system allowing operators to control computer functions by pointing at the computer screen. It consists of a small two-dimensional video sensor, a lightweight hand-held pointer and the OPTIMOUSE electronics package interfacing to a computer in much the same way as a conventional light pen or digitizer tablet. The pointer contains a push button, allowing

menu selection or other forms of data entry. It has been designed for use in environments requiring intensive or tedious data entry or wherever it is inconvenient to use a mouse, digitizing tablet or light pen.

Kantek Inc. manufacture the 3D cordless RingMouse which is placed on the users finger and uses IR and ultrasonic technologies to supply the computer with 3D information. the device comes with an interface box that plugs into the serial port of the computer and tracks the RingMouse. The price for this device is $99.95.

Kinetic Designs Inc. manufactures the MorseK which is a Morse code keyboard emulator program allowing all keyboard keys to be entered via Morse code using any input device connected to any I/O port including the standard keyboard. Some of the features included are: one, two and three-switch modes with user definable delays and switch assignments, user definable audible and visual indicators, a built-in code editor with automatic error detection and a coding scheme using 11 codes for over 60 keys and functions.

Kurta manufactures the XGT which is a graphics tablet input device. The XGT has five different styluses to choose from. The XGT tablet costs $495 and the other necessary equipment is: cordless nonpressure-sensitive pen, $100; cordless 256-level pressure sensitive pen, $200; cordlessfourbutton cursor, $100; and the 16 button cursor, $200.

LC Technologies Inc. manufactures the Eyegaze Computer System which is an eye-operated computer system that enables those with profound physical disabilities to communicate more effectively. The technology uses movement of the human eye to manipulate a personal computer system. The Eyegaze System becomes an eye-operated

command and communication center through which users who have lost motor function can control their living and working environments more efficiently.

Mattel manufacture the Power Glove which is made as a low cost game controller. The Power Glove is made of flexible molded plastic with a Lycra palm. The Power Glove contains resistive-ink flex sensors are located in the back of the fingers which gives the computer finger flex information. This glove is not that accurate, however, it does give a good rough estimate of the hand position.

Newex manufactures the Touchware PC TRANSLATOR which is a mini-console incorporating a touch-sensitive LCD screen. By touching choices on the screen, the user instantly sends information to the PC. Sub-menus provides thousands of additional pre-programmed choices for touch. Customizer software is available to facilitate creating menus.

Pointer Systems Inc. manufactures the FreeWheel Head Pointing System which provides computer access for people whose disabilities prevent them from using their hands on a regular keyboard. An optical pointer provides control of a special cursor, plus menu selection capability. The keyboard appears as a visual image, placed over the screen display. The system allows a person to move a cursor around a monitor screen by using head motion. The reflector may be placed where movement is best with the camera positioned accordingly. The optical camera has a standard input jack which makes it compatible with any stock input switch. The visual keyboard can be dragged to any position on the monitor and can disappear on command. A common word feature completes words to speed up data entry. FreeWheel can be used for environmental control purposes (lights and appliances) with the X-10 Powerhouse. Applications include word

processing, spreadsheets, data communications, desktop publishing, CAD/CAM, and programming.

Spectra Symbol Manufactures membrane switches including the membrane joystick. The membrane joystick is a two DOF finger operated joystick. The user touches the membrane at the desired location and the data is sent to the computer. An advantage of this is the instantaneous position measurement does not have to pass through other points. This advantage is not usable in a telerobitics setting because the manipulator must still pass through the other points that was not required on the membrane. The instant position reading has advantages and disadvantages. A disadvantage is the user is not sure where the manipulator is relative to the membrane. If a user does not have his finger on the membrane and he wants to move the manipulator two inches to the left; where exactly should he or she put the finger. If the finger is placed in the wrong position then the manipulator could be damaged. By lifting the finger and placing to another position the X and Y values change.

Venture Technologies manufactures the TurboSelect which is a keyboard and mouse emulator for people with physical disabilities. TurboSelect replaces the standard input devices (keyboard and mouse) to provide access to a computer. Familiar input techniques including Morse Code, Scanning and Direct Select may be used simultaneously in any of the above techniques to optimize the abilities of the user.

Virtual technologies manufactures the CyberForce and the CyberGlove. The CyberForce supplies grip force-feedback to the user through the CyberGlove. The CyberGlove is an 18 sensor device that monitors the motion of finger bending, roll, pitch, and yaw. There is a 22 sensor Glove that adds a third bend sensor to each finger. Virtual technologies

also manufactures the CyberWear equipment. This equipment, to include the CyberArm, CyberVest, and full-body CyberSuit, measures body motions.

Voice Technologies manufactures the VoiceCAD which is an alternative input device for creating AutoCAD drawings. Voice input simplifies and enhances the input process by allowing the user to focus attention on the drawing rather than on the input procedure.

VPL Research Inc. DataGlove is a patented computer input device which converts hand gestures and positions into computer-readable form. It consists of the DataGlove and a desktop control unit. Sensors mounted on a lightweight lycra glove monitor flexion and extension of the fingers and the position and orientation of the hand. The microprocessor-based control unit acquires data from the DataGlove and transmits it to the host. DataGlove opens up new ways of interacting with computers in CAD/CAM applications, robotics and telemanipulation, simulation and animation.

W Industries manufactures the Space Glove which is use din their Virtuality system. the glove is of hard plastic that fits over the hand. One flex angle is measured for each finger and two flex angles are measured on the thumb. This glove only worked with W Industries products.

C.1.3 Force Feedback Devices

There are currently three companies in the market that have force feedback devices for sale commercially. In the list of address's there are several other companies listed that will supply a force feedback system on a case by case basis. The following company summaries are brief and do very little for the companies. The systems they produce are

simple, complex in their making, and are expensive. These summaries are only to lead the reader into further study if deemed necessary.

Cybernet manufactures the Per-Force Handcontroller comes in two different version. For a complete six DOF system that includes all the hardware to operate, the cost is $60,000. This price includes a 486DX processor, C software library, a demonstration shell, example interfaces, and all documentation. Cybernet demonstrate their device at AFIT and everybody that tried the system was very impressed. A three DOF version is also available to supply a user with linear three DOF force feedback.

EXOS manufacture several input devices one of which is the Sensing and Force reflecting Exoskeleton called SAFIRE. This system, which costs $75,000, provides joint torque feedback to the users fingers and joint torque commands to the slave system. EXOS also manufactures the ArmMaster which supplies the user with five DOF force feedback to the arm. The price of $110,000 also includes a VME controller, A/D, D/A, and a digital signal processor.

EXOS also manufactures several alternate input devices. The Dexterous Hand Master (DHM) is a human hand exoskeleton that measures the human hand and supplies the computer with 20 DOFs. The DHM comes with a PC board, VME, or serial interface. It is adjustable to accommodate many different hand sizes. The price is $15,000. EXOS also makes the Touch master, $4,000; the Position ArmMaster, $22,000; and the Dynamic Wrist Unit (DWU), $5,000.

Sensible Devices Inc. manufactures the PHANToM force-reflecting haptic interface. The PHANToM can be used with a thimble or a stylus to measure and feed back the finger

positions and forces form a slave manipulator. Unlike buzzing tactile simulators the PHANToM supplies actual three DOF feedback to the user's finger tip. The entire system includes the PHANToM, a three channel interface card, and a power interface card for the PC. The price for the entire system is $19,500 which is a relatively good price considering the system that it includes.

Schilling Development manufactures a force feedback manual controller and a kinematics matched manual controller. The force feedback manual controller was originally designed to control the Schilling Titan manipulator. The seven DOF manual controller kenamatically matches the Titan manipulator. The non force feedback manual controller measures the angles of the input device which can be changed to match several different manipulators. The benefit to using a kenamatically matched manual control is the movements seem obvious to the user. If the user moves in a particular direction then manipulator moves in the same direction.

C.2 Survey Analysis Conclusions

This appendix reviewed each of the companies that supply input devices. It showed that there are many companies available that supply input devices. It also showed that there are even more devices that available. There are several companies; however, that should be stressed. CH Products make a very low cost set of devices that may not be the most reliable but are the best for the dollar. Measurements Systems Inc. has a wide veriaty of good reliable devices that will probably supply any users need. Also, there are several companies that are manufacturing alternative devices. With the increase for three DOF devices for Virtual Reality applications, Telerobotics, computer 3D simulations, and some

of the exciting 3D computer games, the input device world has taken on a new outlook. Almost on a daily bases there are new devices available on the market.

APPENDIX D

This appendix lists the addresses and phone numbers of companies that manufacture input devices. It is divided into two tables. The first table lists the companies that manufacture joysticks, mice, and/or trackballs. The first table also lists the companies that manufacture force feedback devices and one-of-a-kind manual controllers for a specific need. The second table lists those companies that manufacture alternate input devices.

Table D.1 Company Addresses

name	address	zip	phone
Advanced Gravis			604-431-5020
Applied Resources Corp.	1275-T Bloomfield Ave Fairfield NJ	07004	201-575-0650
Appoint	1332 Vendels Cir. Paso Robles, CA	93446	805-239-8976
Assmann Data Products	1849 W. Drake Dr., Suite 101 Tempe, AZ	85283	602-897-7001
Bondwell Industrial Co, Inc.	47485-T Sea Bridge Dr. Fremont CA	94538	510-490-4300
CH Products	970 Park Center Dr. Dept TR Vista CA	92083	619-598-2518
CNS Inc.	100 Forde Rd. Denville, NJ	07834	201-625-4056
CTI Electronics Corp.	200 Benton St Stratford CT	06497	203-386-9779
Cyber-tech, Inc.	PO Box 23801 Portland OR	97281- 3801	800-621-8754
GE Co.	3135 Easton Tpke Fairfield CT	06431	800-626-2004
Happ Controls, Inc.	106-T Garlisch Dr. Elk Grove Village IL	60007	708-593-6130
Hed, Hydro Electronics Devices Corp.	PO Box 218 Hartford WI	53027	800-398-2224
Hudson Control Group	44-T Commerce St Springfield NJ	07081	201-376-7400

IBM Corp.	1133 Westchester Ave. White Plains, NY	10604	800-426-9292
Itac Systems Inc.	3121 Benton Drive Garland, TX	75042	800-533-4822
IZU Products Co.	Rt 2 PO Box 3985 Lufkin TX	75903	409-824-3332
Kensington Microware, Ltd.	251 Park Ave. S New York NY	10010	800-535-4242
Kraft Systems Inc	450 W. California Ave Vista CA	92083	619-724-7146
Lexmark International Inc.			
Logitech	6505 Kaiser Dr. Fremont, CA	94555	510-795-8500
Machines Components Corp.	70-T Newtown Rd Plainview NY	11803	800-899-4511
Maurey Instrument Corp.	4557 W. 60th St. Chicago IL	60629	312-581-4555
Measurement Systems, Inc.	777 Commerce Drive Fairfield CT	06430	
Merit, J.R. Controls Inc.	320 Martin Luther King Dr. S. Norwalk CT	06854	800-333-5762
Microsoft Corp.	One Microsoft Way redmond, WA	98052- 6399	206-882-8080
MicroSpeed Inc.	44000 Old Warm Springs Blvd. Fremont, CA	94538	415-490-1403
Mouse Systems Corp.	47505 Seabridge dr. Fremont, CA	94538	415-656-1117
OEM Controls, Inc.	12 Controls Drive Shelton CT	06484	203-929-8431
Orbit Instrument Corp	80-T Cabot Ct Hauppauge NY	11788	
Penny and Giles Controls, Inc.	163 Pleasant St Attleboro MA	02703	
Phase Research	3613-T W MacArthur Blvd Ste 612 NR Santa Anna CA	92704	
Prohance Technologies	1307 S. Mary Ave., #104 Sunnyvale, CA	94087	408-746-0950
PQ Controls Inc	95-T Dolphin Rd Bristol CT	06010	
Quatech Inc.	662-T Wolf Ledges Parkway Akron, OH	44311	
Rexroth Corp. The Industrial Hydraulics Div.	PO Box 2407 2315 City Line Rd Bethlehem, PA	18017	215-694-8300
Sauer-Sundstrand Electronics Systems	3902 Annapolis Lane N Minneapolis, MN	55447	
Sumtak Encoders	615 Pierce St Dept TR Somerset, NJ	08875	908-805-0008
Suncom Technologies	6400 W. Gross Point Rd. Niles, IL	60648	708-647-4040

Sysgration(USA), Inc.	335-T Convention Way Unit D Redwood City, CA	94063	415-306-7860
Z-NIX	211 erie St. Pomona, CA	91768	714-629-8050
Force Feedback Devices			
Cybernet Systems Corp.	919 Green Rd. suite B-101 Ann Arbor, MI.	48105	313-668-2567
Exos Inc.	8 Blanchord Road Burlington MA	01803	617-933-0022
Sensible Devices Inc.	225 Court Street Vanceburg, KY	41179	606-796-6921
System Specific Input devices			
Begej Corp			
Central Research Laboratories Div. Sargent Industries	Hwys. 19 & 16 TK Redwing MN	55066	612-388-3565
Dalmac Inc.	523 Lively Blvd. Elk Grove Village IL	60007	708-364-9262
Honeywell Testing Labs	13350 U.S. Hwy 19 N. Clearwater FL	34624- 7290	813-539-2557
Lamberton Robotics/H.G. Mouat Co. Inc.	P.O Box 127-DN Birmingham AL	35201	800-446-6828
Odetics	1515 S. Manchester Ave. Anaheim, CA	92802	714-774-5000
Orbitec			
Sarcos Inc	261-T E. 300 S Salt Lake City, UT	84111	805-531-0559
Western Space and Marine Inc.	111 Santa Barbara St. Suite T Santa Barbara CA	93101	800-394-3831

Table D.2 Addresses of Companies that Manufacture Alternate Input Devices

name	address	zip	phone
BioControl Systems Inc.	430 Cowper St. Palo Alto, CA	94301	415-329-8494
Carroll Touch Inc.	P.O. Box 1309 Round Rock, TX	78680	512-244-3500
Cirque Corp.	Salt Lake City, UT		800-454-3375
Communication Intelligence Corp.			101-415-7888
ComputAbility Corporation	101 Route 46 East Pine Brook, NJ	07058	800-345-4076
Creative Technolgy Corp.			
Digital Image Design Inc.	170 Claremont Ave., Suite 6 New York, NY	10027	212-222-5236
Global Devices	6630 Arabian Circle Granite Bay, CA	95661	916-791-3533

Greenleaf Medical Systems	2248 Park Blvd. Palo Alto, CA	94306	415-321-6135
GRiD Systems Corp.	47211 Lakeview Blvd. Fremont CA	94537	800-222-4743
Handykey Corp.	141 Mount Sinai Ave. Mount Sinai, NY	11766	800-638-2352
Immersion Human Interface Corp.	P.O. Box 8669 Palo Alto CA	94309- 8669	415-599-5819
Interlink Electronics	Camarillo, CA		805-484-8855
ISCAN Inc.	125 Cambridge Park Dr. P.O. Box 2076 Cambridge, MA	02238	617-868-5353
Kantek Inc.			
Kinetic Designs Inc.	14231 Anatevka Lane SE Olalla, WA	98359	206-857-7924
Kurta	3007 East Chambers St. Phoenix, AZ	85040	800-445-8782
LC Technologies Inc.	4415 Glenn Rose St. Fairfax, VA	22032	703-425-7509
Mattel			
Newex Inc.	100 Drakes Landing Rd. - Suite 260 Greenbrae, CA	94904	415-892-1573
Pointer Systems Inc.	One Mill Street Burlington, VT	05401	800-537-1562
Royal Data Systems	Rt. 14 Box 230Highway 64 West Morganton, NC	28655	800-843-9750
Schilling Development Inc.	1632 DaVinci Court Davis, CA	95616	916-753-6718
Sensible Devices Inc.	225 Court Street Vanceburg KY	41179	606-796-6921
Soricon Corporation	4725 Walnut St. Boulder, CO	18030	800-541-7226
Spectra Symbol Corp.	3101 W. 2100 S Salt Lake City UT	84119	801-972-6995
Venture Technologies Inc.	304 - 134 Abbott Street Vancouver, BC V6B 2K4 Canada		800-663-8931
Virtual Technologies	2175 4th Ave., Suite 510 S.W. Calgary, Alberta, Canada		
Voice Technologies	120 Village Square - #143 Orinda, CA	94563	415-283-7586
VPL Research Inc.	656 Blair Island Rd. - Suite 304 Redwood City, CA	94063	415-361-1710
W Industries			

APPENDIX E

This appendix will discuss the CHIMERA real time operating system. It will develop an entire CHIMERA program. It will start with the *config* files, then discuss the main program, and finally, how to make a module using *modmaker*. It will also discuss lessons learned while programming the interface modules. This appendix will also discuss the modules that were made during this research.

E.1 Introduction to CHIMERA

This section will discuss the preliminary steps necessary to get a CHIMERA program running. It will also gives a brief background into CHIMERA and how it operates.

CHIMERA is a concept and product form Carnegie Mellon University (CMU). They designed this product as a real time operating system which runs on a VMEbus and uses a Sun workstation for program development. CHIMERA has been distributed to several universities and government installations. The organizations involved with the Air Force generic telerobotics architecture currently have a running copy of CHIMERA. When this thesis project is complete, all participating organizations will get a copy of this work and will be able to run the code almost instantly.

E.1.1 Basics of CHIMERA

It is assumed that CHIMERA has been installed properly on a VME/Sun system and is operating properly. If there are any installation problems, then the CHIMERA manual

should be referenced (26:269). Once the CHIMERA operating system has been installed, then the proper paths should be set into your environment. The proper setting for the environment variables are contatined in the CHIMERA manual (26:2).

CHIMERA uses a state table and memory sharing as its main premise for handling time critical variables. The state table is created by the main program and modified by modules and/or the main program depending on the particular requirements. CHIMERA also handles the timing of the modules through the files called rmod files. The rmod files are a set of files that contain a set of Reconfigurable variables for the respective MODule (thus the name RMOD). An example system using CHIMERA is as follows. A state table is generated using the file called main.c and several modules are linked to the main. The state table consists of 12 joint variables to operate a PUMA robot. The variables are six joint reference variables and six joint measured variables. Lets say there are two modules, one to command the joint reference variables and the other to control the PUMA robot. The first module will simply command the appropriate joint variables to the desired value by writing to the state table. The second module then reads those same variables and commands the PUMA robot to the desired value using the measured variables to close the feedback loop. The actual PUMA commands will not be discussed in this report because they are hardware dependent and beyond software implementation.

E.1.2 Starting From Scratch

The first order of business to any set of code is the proper modeling of the requirements. The program developer should set down and design a complete set of system requirements. The programmer should then decide how best to utilize CHIMERA to

handle the problem requirements. Once the program has been laid out, then the CHIMERA programming can begin.

To construct a complete CHIMERA program, there must be two configuration files, a main program, and one or more modules. These files should be placed in the proper directories as described in the CHIMERA manual (26:26).

The first file that must be constructed is the state variable table configuration file. This file tells CHIMERA how many state variables there are for a particular program and define those variables. The following is an example file:

```
# *********************************************************************
    #
#                                                                  #
#   puma.svar                                                      #
#                                                                  #
# ------------------------------------------------------------     #
#                                                                  #
#       This file provides state variable table information for #
#       operating a PUMA 560.                                      #
#                                                                  #
#                                                                  #
# *********************************************************************
    #

# Reference variables                                              #
# Q_REF                 : Reference joint positions                #
NAME  Q_REF
TYPE  float
DESC  reference joint position
UNITS  radians
NELEM  6
MIN   -2.79  -3.93  -0.89  -1.92  -1.75  -4.71
MAX    2.79   0.70   4.05   2.97   1.75   4.71

# Measured variables                                               #
```

```
# Q_MEZ                  : Joint positions as read from position encoders         #

NAME   Q_MEZ
TYPE   float
DESC   measured joint position
UNITS  radians
NELEM  6
MIN    -2.79  -3.93  -0.89  -1.92  -1.75  -4.71
MAX     2.79   0.70   4.05   2.97   1.75   4.71

EOF
```

The second file needed is the subsystem file. The subsystem file defines the hardware being used in the VME system. In this example, the VME chassis has two processors, one called *control* and the other called *crusher*. The following is an example file called puma.sbs:

```
# *********************************************************************#
#                                                                      #
#       puma.sbs                                                       #
#                                                                      #
#              created by            Tom E Deeter       01-08-94       #
#                                                                      #
# ----------------------------------------------------------------     #
#                                                                      #
#       This file provides subsystem configuration information         #
#       for control of the AFIT PUMA 560 robot.                        #
#                                                                      #
# *********************************************************************#
# Subsystem information                                                #
# SUBSYSTEM:   Name of subsystem                                       #
# SVARFILE:    Name of file for subsystem's state variable table       #
# MASTER:      Name of RTPU or memory board to use for master's IPC    #
#              segments. The svartable will also be stored on this     #
#              RTPU. (optional, default is 'RTPUNAME(getbid())'.)      #
SUBSYSTEM    puma
SVARFILE     puma.svar
MASTER       control
# RTPU information                                                     #
# RTPU:           Name of RTPUs used by subsystem                      #
RTPU         control
```

```
RTPU        crusher
EOF
```

The next required file is the main program. The main program can be used to control all developed modules by spawning processes and killing processes. CHIMERA will also turn the modules on and off when required by the system. CHIMERA has a developmental mode that allows the user to spawn, start, stop, and kill processes. The details of both modes are in the CHIMERA manual (26:214-231). The following code shows the file called puma.c which is the main program for this example. This file causes CHIMERA to be used in the developmental mode of programming; thus, the user must manually spawn, start, stop, and kill the modules.

```
/* ************************************************************************/
/*                                                                      */
/*      puma.c                                                          */
/*                                                                      */
/*              created by          Tom E. Deeter             01-08-94  */
/*                                                                      */
/* ---------------------------------------------------------------      */
/*                                                                      */
/*      Main routine for driving the AFIT PUMA 560 manipulator.         */
/*                                                                      */
/* ************************************************************************/


/* ************************************************************************/
/*      include files                                                   */
/* ************************************************************************/

#include <chimera.h>
#include <sbs.h>


#define MASTERRTPU "control"

int     main()
{
```

```
 sbsSystem_t  *sbs;
 sbsTask_t    **taskptr;
 char         *rtpu;
 char         *module;

 sbsServer();

  if (strcmp(RTPUNAME(getbid()),MASTERRTPU) == 0)
 {
  sbs = sbsInit("tmain");

/* Code to spawn, start, stop, and kill modules goes here */

  sbsCmdi(sbs, NULL);  /* used to enable module developement    */
  sbsFinish(sbs);      /* stop chimera operating system         */
 }

 return 0;
}
```

To compile the main program, it is highly recommended to use a makefile. A complete description of using makefiles is given in the CHIMERA manual(26:15,16). The last set of code is the modules themselves. The modules can be written manually or the program *modmaker* can be used. Modmaker will interactively build a module. In this example, there are two modules required. One to take the user input and direct the Q_REF variables accordingly. The other module will take those Q_REF variables, subtract the Q_MEZ variable and send the resultant to the appropriate place to command the puma to move. The module to read in the users input and change the Q_REF variables is below:

```
/* ***********************************************************************/
/*                                                                       */
/*      trjjgen.c                                                        */
/*                                                                       */
/*              created by          Wayne F. Carriker     04-21-93       */
/*                                                                       */
/*                      Carnegie Mellon University                       */
/*                                                                       */
/*              modified by         Wayne F. Carriker     06-23-93       */
```

```
/*                                                              */
/*              fixed up for change in Chimera 3.0              */
/*              involving reading OUTVARS in xxxOn and          */
/*              cleaned up before review                        */
/*                                                              */
/*      reviewed by        $ Some name here $   $ date $        */
/*                                                              */
/* ------------------------------------------------------       */
/*                                                              */
/*   Online joint space trajectory generation module            */
/*                                                              */
/*   State variable table:                                      */
/*      INCONST:       NDOF - number of degrees of freedom      */
/*      OUTCONST:      none                                     */
/*                                                              */
/*      INVAR:         Q_MEZ - measured joint positions         */
/*      OUTVAR:        Q_REF - reference joint positions        */
/*                                                              */
/*   Special notes:                                             */
/*      This module is based on code created by Richard         */
/*   Volpe.  This module is limited to trajectories for upto    */
/*   MAXJOINTS joints.                                          */
/*                                                              */
/* *************************************************************************/


/* *************************************************************************/
/*   include files                                              */
/* *************************************************************************/

#include <chimera.h>
#include <sbs.h>
#include <limits.h>
#include <math.h>
#include <string.h>
#include <cmdi.h>
#include <ui.h>


/* *************************************************************************/
/*   macro definitions                                          */
/* *************************************************************************/

#defineMAXJOINTS 10
```

```
#define QEPSILON     0.001

#define MIN_DURATION          1.0
#define MAX_DURATION          120.0
#define DEF_DURATION          10.0

#define LINEAR                1
#define CYCLOID               2
#define FIFTH_ORDER_POL       3

#define DUMMY_CODE            1


/* ********************************************************************/
/*      local function prototypes                                    */
/* ********************************************************************/

static double   profilefunc(double t, int type);


/* ********************************************************************/
/*      module 'Local_t' definition as required by Chimera           */
/* ********************************************************************/

typedef struct {
 int            *Ndof;
 float          *Qmez, *Qref;
 int            stepnum, type;
 double         duration, stepsize;
 double         qdelta[MAXJOINTS], qinit[MAXJOINTS];
} trjjgenLocal_t;

SBS_MODULE(trjjgen);

/* ********************************************************************/
/*      functions                                                    */
/* ********************************************************************/


/* ********************************************************************/
/*      trjjgenInit                  Initialize the module.          */
/* ********************************************************************/

int     trjjgenInit(cinfo, local, stask)
```

```
cfigInfo_t       *cinfo;
trjjgenLocal_t *local;
sbsTask_t        *stask;
{
  sbsSvar_t               *svar = &stask->svar;


  /* Get pointers to state variables.                              */

  local->Ndof = svarTranslateValue(svar->vartable, "NDOF", int);
  local->Qmez = svarTranslateValue(svar->vartable, "Q_MEZ", float);
  local->Qref = svarTranslateValue(svar->vartable, "Q_REF", float);


  /* Ensure that NDOF <= MAXJOINTS.                                          */

  if (*(local->Ndof) > MAXJOINTS)
  {
    printf("Module `trjjgen' can only handle upto %d ", MAXJOINTS);
    printf("degrees of freedom: NDOF = %d\n", local->Ndof);
    errInvoke(stask->errmod, "NDOF is too large", DUMMY_CODE);
  }


  /* Return from initialization.                                   */

  return (int) local;
}


/* ******************************************************************* */
/*       trjjgenOn                      Start up the module.           */
/* ******************************************************************* */

int      trjjgenOn(local, stask)
trjjgenLocal_t *local;
sbsTask_t        *stask;
{
  char           *head, *tail;
  int            flag = 0, i, n, *ptr3;
  float          *qref = local->Qref, *ptr1;
  double         dummy[MAXJOINTS], *ptr2;
  double         *qdelta = local->qdelta, *qinit = local->qinit;
  UI             *ui;
  svarVar_t      *svarQref;
```

```
sbsSvar_t      *svar = &stask->svar;


/* Get a pointer to the state variable table for user I/O.          */

svarQref = svarTranslate(svar->vartable, "Q_REF");


/* Store the initial joint positions.                               */

n = *(local->Ndof);
for (i = 0; i < n; ++i)
  qinit[i] = (double) qref[i];


/* Check for command line inputs.                                   */

if (strlen(stask->argptr) > 0)
{
  flag = 1;


  /* Try to get NDOF joint positions first.                         */

  head = stask->argptr;
  for (i = 0; i < n; ++i)
  {
    dummy[i] = cmdiArgDouble(&head, &tail, NULL, MAXDOUBLE);
    if (fabs(MAXDOUBLE - dummy[i]) < 1.0)
      break;
    head = tail;
  }


  /* If all the joint positions were available try to get the      */
  /* trajectory duration and type.                                  */

  if (i != n)
    flag = 0;
  else
  {
    local->duration = cmdiArgDouble(&head, &tail, NULL, MAXDOUBLE);
    head = tail;
```

```
  local->type = cmdiArgInt(&head, &tail, NULL, INT_MAX);

  if (local->type == INT_MAX)
    flag = 0;
}


/* If anything was missing from the command line report `short'.  */

if (!flag)
  printf("Too few command line parameters\n");


/* If there is still more command line left report `long'.          */

if (flag && strlen(tail) > 0)
{
  flag = 0;
  printf("Too many command line parameters\n");
}


/* If there have been no problems, ensure that all the data values  */
/* are in range.                                                     */

if (flag)
{
  for (i = 0; i < n; ++i)
  {
    if (dummy[i] < svarMin(svarQref, float)[i])
      flag = 0;
    if (dummy[i] > svarMax(svarQref, float)[i])
      flag = 0;
  }
  if (local->duration < MIN_DURATION || local->duration > MAX_DURATION)
    flag = 0;
  if (local->type < LINEAR || local->type > FIFTH_ORDER_POL)
    flag = 0;


  /* If anything was out of range, report the problem.              */

  if (!flag)
    printf("Command line data out of range\n");
```

```
  }
}

/* If no information was provided on the command line, or if that  */
/* information was wrong, ask user for trajectory information.      */

if (!flag)
{
 ui = uiCreate(3, 512);

 /* Get the joint information first.  Use floats because all the      */
 /* state variable table variables are floats and convert to double   */
 /* when putting the values into `dummy'.                             */

 ptr1 = (float *) uiVector(ui, "Final joint position", VT_FLOAT, n,
                  local->Qref, svarMin(svarQref, float),
                  svarMax(svarQref, float));

 /* Get the trajectory duration next.                     */

 ptr2 = uiDouble(ui, "Trajectory duration",
                 DEF_DURATION, MIN_DURATION, MAX_DURATION);

 /* Get the trajectory type last.                              */

 ptr3 = uiInt(ui, "Trajectory type",
              FIFTH_ORDER_POL, LINEAR, FIFTH_ORDER_POL);

 /* Call the user interface.                                          */

 sbsUserInput(stask, ui);

 /* Copy the data into the appropriate locations.              */

 for (i = 0; i < n; ++i)
   dummy[i] = (double) ptr1[i];

 local->duration = *ptr2;

 local->type = *ptr3;
```

```
  uiFree(ui);
 }

 /* Set remaining parameters.                                          */

 for (i = 0; i < n; ++i)
   qdelta[i] = dummy[i] - qinit[i];

 local->stepnum = 0;
 local->stepsize = 1.0 / (stask->freq * local->duration);

 /* Return from start up.                                              */

 return I_OK;
}

/* ******************************************************************** */
/*       trjjgenCycle              Process module information.         */
/* ******************************************************************** */

int       trjjgenCycle(local, stask)
trjjgenLocal_t  *local;
sbsTask_t       *stask;
{
 int            flag, i, n = *(local->Ndof), type = local->type;
 float          *qmez = local->Qmez, *qref = local->Qref;
 double         nt, temp;
 double         *qdelta = local->qdelta, *qinit = local->qinit;

 /* Calculate the next set of joint positions.                 */

 if ((nt = local->stepnum++ * local->stepsize) > 1.0)
   temp = 1.0;
 else
   temp = profilefunc(nt, type);

 for (i = 0; i < n; ++i)
   *(qref++) = (float) (*(qinit++) + temp * *(qdelta++));
```

```
 /* See if it is time to stop.          Ideally, when nt == 1.0, the robot       */
 /* has reached the desired position, but just to be sure, check the   */
 /* actual position against the desired position before turning off.   */

 if (nt > 1.0)
 {
  qref = local->Qref;
  flag = 1;
  for (i = 0; i < n; ++i)
   if (fabs(*(qref++) - *(qmez++)) > QEPSILON)
    flag = 0;

  if (flag)
   return SBS_OFF;
 }


 /* Return at the end of each cycle.                              */

 return I_OK;
}


/* *********************************************************************/
/*      trjjgenOff                   Stop the module.                  */
/* *********************************************************************/

int      trjjgenOff(local, stask)
trjjgenLocal_t *local;
sbsTask_t      *stask;
{
 /* Indicate that the module is off and return.                */

 printf("trjjgen: OFF\n");
 return I_OK;
}


/* *********************************************************************/
/*      trjjgenKill                  Clean up after the module.        */
/* *********************************************************************/

int      trjjgenKill(local, stask)
```

```
trjjgenLocal_t  *local;
sbsTask_t       *stask;
{

 /* Indicate that the module is finished and return.                    */

 printf("trjjgen: FINISHED\n");
 return I_OK;
}


/* ********************************************************************/
/*      trjjgenError                Attempt automatic error recovery.    */
/* ********************************************************************/

int     trjjgenError(local, stask, mptr, errmsg, errcode)
trjjgenLocal_t  *local;
sbsTask_t       *stask;
errModule_t     mptr;
char            *errmsg;
int             errcode;
{
 /* Return after not correcting error.                                  */

 return I_OK;
}


/* ********************************************************************/
/*      trjjgenClear                Clear error state of the module.     */
/* ********************************************************************/

int     trjjgenClear(local, stask, mptr, errmsg, errcode)
trjjgenLocal_t  *local;
sbsTask_t       *stask;
errModule_t     mptr;
char            *errmsg;
int             errcode;
{
 /* Return after not clearing error.                                    */

 sbsNewError(stask, "Clear not defined, still in error state", errcode);
 return SBS_ERROR;
}
```

```
/* ******************************************************************/
/*      trjjgenSet                    Set module parameters.         */
/* ******************************************************************/

int     trjjgenSet(local, stask)
trjjgenLocal_t *local;
sbsTask_t      *stask;
{
  return I_OK;
}

/* ******************************************************************/
/*      trjjgenGet                    Get module parameters.         */
/* ******************************************************************/

int     trjjgenGet(local, stask)
trjjgenLocal_t *local;
sbsTask_t      *stask;
{
  return I_OK;
}

/*******************************************************************/
/*    trjjgenReinit                                                  */
/*******************************************************************/

int     trjjgenReinit(local, stask)
trjjgenLocal_t *local;
sbsTask_t      *stask;
{
  return I_OK;
}

/*******************************************************************/
/*      trjjgenSync                                                  */
/*******************************************************************/

int     trjjgenSync(local, stask)
trjjgenLocal_t *local;
sbsTask_t      *stask;
{
```

```
  return I_OK;
}


/* ***********************************************************************/
/*     profilefunc            Choose trajectory profile type.            */
/* ***********************************************************************/

static double   profilefunc(double t, int type)
{
  static double  twopi = 6.283185308, twopiinv = 0.1591549431;
  double        p;


  switch(type)
  {
    case LINEAR:
      p = t;
      break;
    case CYCLOID:
      p = t - sin(twopi*t) * twopiinv;
      break;
    case FIFTH_ORDER_POL:
    default:
      p = t*t*t*(10.0 - 15.0*t + 6.0*t*t);
      break;
  }
  return p;
}
```

The example module shows the user interface commands as well as how to implement a function in a module. The second module is hardware specific; thus, will not be discussed.

E.1.3 Modules Developed

There were several modules developed during this research which were used to interface CHIMERA to the user input devices. The modules developed were modules to interface

the Thrustmaster joystick, Logitech mouse, Schilling master control unit, and a DIM6 spaceball.

There were two modules constructed for the Thrustmaster joystick. The first module operated the PUMA in joint space while the second module operated the PUMA in Cartesian space. It was quickly learned that the Cartesian space control was the most desirable of the two modes.

The Logitech mouse was also interfaced to CHIMERA and operated the PUMA in Cartesian space. The module controlled the PUMA in three DOF by using the three buttons on the mouse. While the left button was pressed, the PUMA would move in the Y, Z plane and when the middle button was pressed the mouse would move the PUMA in the X, Z plane and finally while holding the right mouse button the PUMA would move in the X, Y plane. This type of operation worked well until a three DOF task was required in real time. To overcome this deficiency, shared control was used and the task was accomplished.

The Schilling master control unit was also interfaced to CHIMERA. The Schilling master control unit at AFIT was made to control the HYDRA which is a fuel tank deseal/seal system. The sensor was a six DOF unit that did not kenamatically match the PUMA manipulator. This caused considerable confusion for the user, since the joint motion did not match the joint motion of the PUMA.

The last module developed to interface input devices was a module to interface the DIM6 spaceball to CHIMERA. The DIM 6 is a six DOF input device that uses force sensors to direct motion of the PUMA. It was found that using the DIM 6 in its full six DOF mode

was extremely confusion and cumbersome. If the DIM 6 was used in its three DOF mode, the PUMA could be moved rather gracefully. The DIM 6 has a button that allows the user to input only translation inputs and another button to allow the user to input rotational inputs. The separation of rotational and transitional inputs was a welcome feature for the user.

E.2 Conclusion

This appendix gave some incite into constructing a CHIMERA run program. It showed how to start from scratch and make a complete CHIMERA program to run a puma robot. This appendix also covered the modules developed for this research.

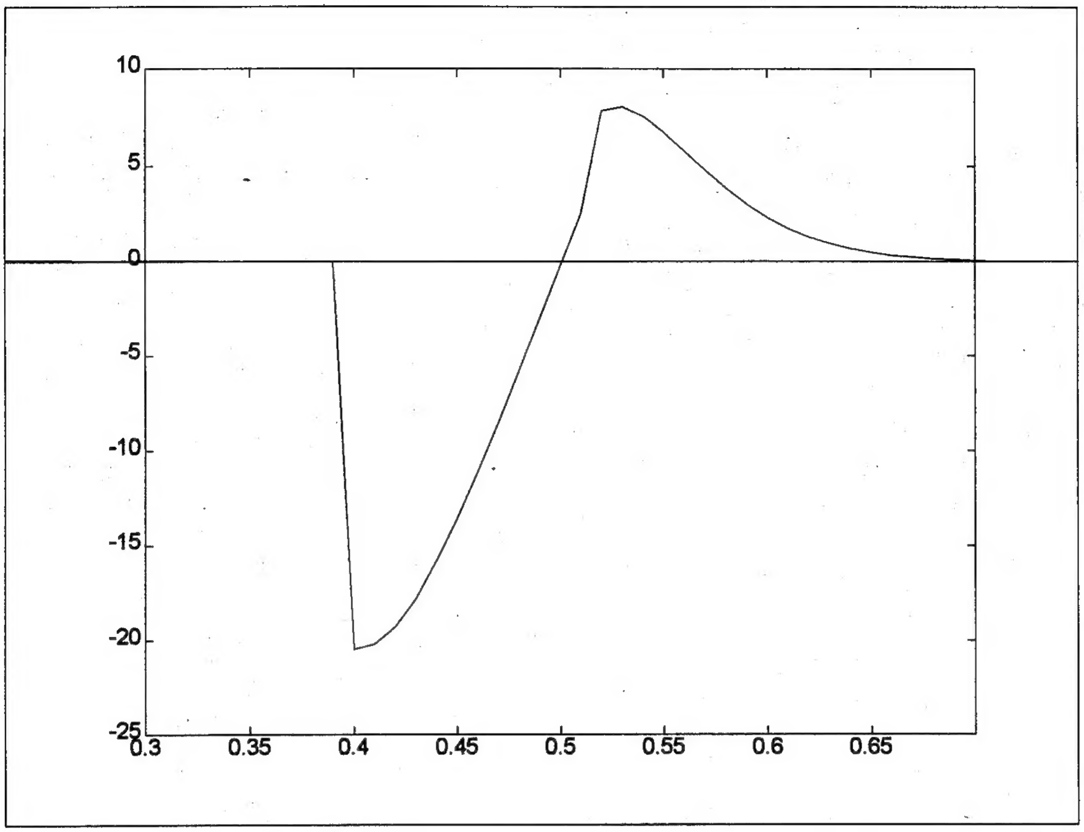
APPENDIX F

F.1 Hardware Used for Research

The hardware used for this research was the PUMA 560 robot that had the control circuitry replaced with a TRIDENT board. Also used was a VME chassis using CHIMERA as the operating system and a Sun Sparc2 workstation.

While using the hardware, a serious problem was discovered with the TRIDENT board. At the time, it was not known if the problem was in the PUMA or the TRIDENT board. After recording the reference and measured positions of the robot, it was discovered that the problem was in the measured side of the control system (see Figure F.1). The figure shows the measured value suddenly changing from 20 degrees and back to 0 degrees with no commanded change. The next step to trouble analysis was to isolate the component on the measured side that was causing the problem. To aid in this procedure, I asked for the help of a digital engineering lab instructor Lt. Col Wailes, who loaned a logic analyzer from the AFIT digital laboratory to the robotics lab for this particular problem.

Figure F.1 Joint Error Plot



Using the logic analyzer, I discovered that the problem occurred when joint six encoders where being read by the processor. When joint six encoders were read, the ground level for joints five and six increased by about 3.2 volts. This ground level fluctuation caused the joint five most significant bit encoder to reset or go to zero. The most significant bit caused about a 20 degree step response to the joint five motor which corresponds to the error plot shown in Figure F.1. The designer of the board was then called to see if it could be fixed. The designer recommended an extra ground wire be added for the board ground

to the joint five and six ground. This solution reduced the ground fluctuation to under one volt. The puma has not had any problems since.

Also used during this research was a VME chassis with CHIMERA as the operating system. The chassis has two 68030 processors, four IO boards, a JR3 force sensor board, and an A/D board. The only external board used during this research was the JR3 force sensor board. The setup and code to operate the sensor was supplied by Carnegie Mellon University. The serial ports on the two processors were used as the hardware connections between the input devices and the system. CHIMERA has several IO drivers that were used to command the serial port. The Silicon Graphics Indigo was used to display a robotics simulation package. During code development and PUMA problems, the simulation was used to debug the code. A socket was established between CHIMERA and the host Sun and from the host Sun to the SGI. The socket data was then read in by the simulation package and the PUMA displayed appropriately.

F.2 Conclusion

During this research hardware was repaired and used to demonstrate the use of the input devices. This thesis work required the use of hardware. This was extremely time consuming and also gratifying. Time consuming due to the troubles and gratifying to see the end result in actual manipulator motion.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 1994

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
Survey and Implementation of Commercial Manual Controllers for a Generic Telerobotics Architecture

5. FUNDING NUMBERS

6. AUTHOR(S)
Thomas E. Deeter

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/94D-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Capt Brian Cassiday
SA-ALC/TIEST Bldg 183
450 Quentin Roosevelt Rd.
Kelly AFB, TX 784241-6416

Capt Paul Whalen
AL/CFBA Bldg 441
2610 Seventh St.
WPAFB, OH 45433-6583

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)* The purpose of this study is to determine an input device for the Air Force's generic telerobotics architecture for large aircraft maintenance and repair. One area of concern is the human to machine interface, more specifically, which manual controller should be used for the specified tasks in this architecture. We mailed a survey to 68 companies in order to compile a list of possible input devices that the telerobotics architecture could use. 32 companies responded which gave me enough data to generate a list that described the physical traits of the input devices. We then divided the required tasks into actions and analyzed them to generate a list of traits required by an input device. Both the task analysis and device listings were combined mathematically to form a performance table which revealed the possible devices that could perform each individual action. To aid in development of the Air Force's generic telerobotics architecture, we integrated four input devices into a VME based operating system called CHIMERA. These four devices represent the four different sensor types that are currently available in today's market. The first device is a mouse which relays position changes of the mouse to the computer. The second device is a joystick that can be used in two different ways. The joystick can measure position data of the hand position or it can measure the displacement of the hand from the center of the total movement. The third device is a six degree-of-freedom (DOF) spaceball that measures the amount of force for position data and rotational data. The fourth device is a Schilling manual controller which has a one-to-one mapping from the controller joints to the manipulator joints.

14. SUBJECT TERMS
Manual Controller, Input Device, Aircraft Maintenance, Shared Control, Telerobotics

15. NUMBER OF PAGES
128

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std Z39-18
298-102